Algorithms for Source Separation - with Cocktail Party Applications

Olsson, Rasmus Kongsgaard

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Algorithms for Source Separation - with Cocktail Party Applications

Rasmus Kongsgaard Olsson

Kongens Lyngby, 2007
IMM-PHD-2006-181

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

In this thesis, a number of possible solutions to source separation are suggested. Although they differ significantly in shape and intent, they share a heavy reliance on prior domain knowledge. Most of the developed algorithms are intended for speech applications, and hence, structural features of speech have been incorporated.

Single-channel separation of speech is a particularly challenging signal processing task, where the purpose is to extract a number of speech signals from a single observed mixture. I present a few methods to obtain separation, which rely on the sparsity and structure of speech in a time-frequency representation. My own contributions are based on learning dictionaries for each speaker separately and subsequently applying a concatenation of these dictionaries to separate a mixture. Sparse decompositions required for the decomposition are computed using non-negative matrix factorization as well as basis pursuit.

In my work on the multi-channel problem, I have focused on convolutive mixtures, which is the appropriate model in acoustic setups. We have been successful in incorporating a harmonic speech model into a greater probabilistic formulation. Furthermore, we have presented several learning schemes for the parameters of such models, more specifically, the expectation-maximization (EM) algorithm and stochastic and Newton-type gradient optimization.

Resumé

Jeg foreslår i afhandlingen en række løsninger på kildeseparationsproblemet. Metoderne er væsensforskellige, men har det til fælles, at de i høj grad er afhængige af problemspecifik viden. Flertallet af algoritmerne er udviklet med henblik på taleanvendelser, og netop derfor er strukturelle egenskaber ved tale blevet indbygget.

Enkeltkanalseparation af tale er en særlig krævende signalbehandlingsdisciplin, hvor formålet er at udtrække en række talesignaler fra et enkelt observeret mikstursignal. Jeg præsenterer en række separationsmetoder, som udnytter tales meget spredte fordeling i en tids-frekvens-repræsentation. Mine egne bidrag er baseret på at lære 'ordbøger' for hver enkelt taler, som senere kan bruges til at adskille signalerne. Matrixfaktorisering og basis pursuit bruges til at beregne dekompositionerne.

I forbindelse med mit arbejde med fler-kanalproblemet, har jeg koncentreret mig om foldningsmiksturer, som er en passende model i akustiske problemer. Det er lykkedes os at indbygge en harmonisk talemodel i en sandsynlighedsteoretisk ramme. Desuden har vi præsenteret flere fremgangsmåder til indlæring af modellens parametre. Mere specifikt, har vi benyttet EM algoritmen, stokastisk gradient samt en Newton-forbedret gradientmetode.

Preface

The preparation of a thesis is one of the requirements to obtain a Ph.D. degree from the Technical University of Denmark (DTU). The main objective is to put into context the research conducted and published in my three years as a Ph.D. student. I do not repeat the narrative flows of the articles, but rather, I introduce the field, citing the relevant literature. Below, I have listed the published works, the roman numeral identifying their location in the appendix.
In the field of single-channel separation:

I B. A. Pearlmutter and R. K. Olsson, Algorithmic Differentiation of Linear Programs for Single-channel Source Separation, in proceedings of IEEE International Workshop on Machine Learning and Signal Processing, 2006

II M. N. Schmidt and R. K. Olsson, Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization, in proceedings of International Conference on Spoken Language Processing, 2006

III M. N. Schmidt and R. K. Olsson, Feature Space Reconstruction for Single-Channel Speech Separation, in submission to Workshop on Applications of Signal Processing to Audio and Acoustics, 2007

- H. Asari, R. K. Olsson, B. A. Pearlmutter and A. M. Zador, Sparsification for Monaural Source Separation, in Blind Speech Separation, eds. H. Sawada, S. Araki and S. Makino, Springer, 2007 - in press

In the field of multi-channel separation:

IV R. K. Olsson and L. K. Hansen, Probabilistic Blind Deconvolution of Non-stationary Sources, in proceedings of European Signal Processing Conference, 1697-1700, 2004

V R. K. Olsson and L. K. Hansen, Estimating the Number of Sources in a Noisy Convolutive Mixture using BIC, in proceedings of International Conference on Independent Component Analysis and Blind Signal Separation, 618-625, 2004

VI R. K. Olsson and L. K. Hansen, A Harmonic Excitation State-Space Approach to Blind Separation of Speech, in Advances in Neural Information Processing Systems, 17, eds. L. K. Saul, Y. Weiss and L. Bottou, MIT Press, 993-1000, 2005

VII R. K. Olsson, K. B. Petersen and T. Lehn-Schiøler, State-Space Models - from the EM algorithm to a Gradient Approach, Neural Computation, 19(4), 1097-1111, 2007

VIII R. K. Olsson and L. K. Hansen, Linear State-space Models for Blind Source Separation, Journal of Machine Learning Research, 7, 2585-2602, 2006

IX R. K. Olsson and L. K. Hansen, Blind Separation of More Sources than Sensors in Convolutive Mixtures, International Conference on Acoustics on Speech and Signal Processing, 5, 657-660, 2006

Acknowledgements

Having spent three years at the Intelligent Signal Processing group, it seems that positive memories dominate. There was no shortage of enthusiastic people willing to engage in research collaboration, I was fortunate to publish joint papers with fellow students Tue Lehn-Schiøler, Kaare Brandt Petersen and Mikkel Schmidt as well as my supervisor Lars Kai Hansen. In addition, I enjoyed the sometimes heated lunch-time discussions about everything and nothing.

I owe a special thanks to my supervisor Lars Kai Hansen, always competent, and serving as an inspirational force, providing pep-talks in times of need.

On many levels it was an enriching experience to spend time at Hamilton Institute in Maynooth (nearby Dublin). I was very well received by Barak Pearlmutter who supervised my research in this period of time. I had a really good time with nice people coming to booming Ireland from all over the world. Most memorably, I played in the institution's soccer team, which gloriously beat the biology department team on at least two occasions.

I wish to thank the proofreaders Tariq Khan, Jakob Kongsgaard Olsson and Mikkel Schmidt.

The stipend awarded by Oticon Fonden made this work possible.

Contents

Chapter 1

Introduction

It is a non-negotiable condition of agents operating in the real world that the environment is observed only through sensors, obscuring the objects of interest. Humans are equipped with advanced sensory devices and formidable processing which partially alleviate this limitation. Imagine, for instance, that you are attending a cocktail party, listening to your friend speaking. Depending on the conditions, you are able isolate your friend's speech and recognize the words at little effort, despite a number of interfering voices and other sounds (Cherry, 1953) . This is a clear indication that the human auditory system has a mechanism for separating incoming signals, and indeed, much research has been directed at describing the psychoacoustics more closely (Bregman, 1990). Not only are we interesting in employing a machine to emulate human auditory perception, more generally, we hope to devise algorithms that can extract hidden source signals in a large range of sensory domains. Applications range from automatic speech recognition to analysis of brain images.

This thesis is concerned with constructing algorithms for source separation, thus rendering it possible to treat the cocktail party problem and related scenarios. More generally, source separation is a relevant procedure in cases when a set of source signals of interest has gone through a unspecified mixing process and has been recorded at a sensor array. Given the observed mixture signals, the objective is to invert the unknown mixing process and estimate the source signal (Figure 1.1). In many cases, this is possible, even when placing only limited as-

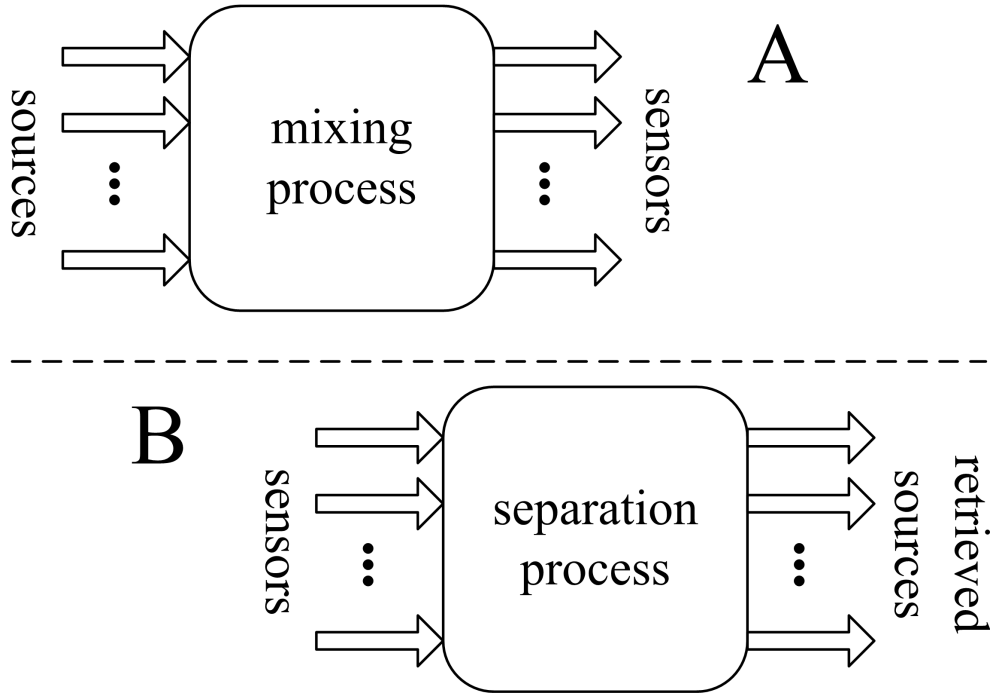

Figure 1.1: Conceptualized visualization of the source separation problem. In **A**, we are presented with the premise: the source signals of interest are observed only through the sensors. It is the end goal to retrieve the sources as illustrated in **B**. The mixing process is often unknown and has to be estimated as part of the procedure. Whether the problem can be solved depends on the properties of the mixing process and the sources.

sumptions on the mixing process, such as linearity. The main method for such a relatively general attempt at source separation is independent component analysis (ICA, Hyvärinen et al., 2001). However, a truly general solution to source separation does not exist, *e.g.*, the mixing mapping may be non-invertible.[1] Hence, the solution often have to be tailored to the problem at hand.

Thus, rather than committing to developing a canonical method, the focus of the thesis is on combining machine learning techniques and expert knowledge specific to the data domain: flexible, data-driven signal models are infused with

[1]Consider for example the sum of two i.i.d. Gaussian sources, $y = s_1 + s_2$, where s_1 and s_2 are Gaussian stochastic variables. Hence, the mixture, y, is also i.i.d. Gaussian. By observing y, we can estimate its mean and variance, each the sum of the means and variances s_1 and s_2. As a result, the individual statistics of s_1 and s_2 are unavailable due to the fact that a continuum of solutions exists. Hence, the sources, s_1 and s_2, cannot be inferred from the observable.

all available knowledge. We often know which types of sources to expect being present in the mixture and how they are mapped into the mixture.

There are more ways to make the fullest use of priori knowledge in machine learning. The first way is to simply make clever choices regarding the representation of data. For instance, many natural sounds reveal patterns of interest once they are mapped to a time-frequency representation. Furthermore, mixed signals may decompose in the transformed domain. In computational auditory scene analysis (CASA), algorithm researchers and designers seek to copy the internal representations of the human auditory system. CASA builds on detailed knowledge of psychoacoustics, ranging from the pinna to neuronal processing. Bregman (1990) described how humans perform ASA (and thus, source separation) by employing cues appearing along the auditory pathway to group and segregate the fragments of the various audio sources.

The second way consists in formulating a generative model of the problem which can be a full physical model, or, at least incorporate the relevant structure of the signals into probability density functions. This naturally applies to speech, where hidden Markov models (HMM) and sinusoidal models are widely used in automatic speech recognition as well as in efficient coding (Rabiner, 1989; McAulay and Quateri, 1986). It is a sensible hypothesis, and one that is empirically justified, that source separation can benefit from sensible preprocessing as well as detailed speech modelling. In fact, combining the two approaches to knowledge inclusion is a key objective of my research.

1.1 Organization

The thesis is mainly structured according to a specific property of the problem, namely the number of sensors, Q, available to the source separation algorithm. Single-channel separation ($Q = 1$), which is the hardest case, is treated in chapter 2. Assuming additive mixing, the problem is to estimate the source signals from the sum alone.[2] This is only possible when the sources are sufficiently structured in time, or, trivially, defined on separate intervals. Typically the difficulty of the

[2]That is, estimate s_i from $y = \sum_{i=1}^{P} s_i$, where P is the number of sources.

problem is dramatically reduced when more sensors are allowed ($Q > 1$). In multi-channel separation, fairly general tools can be used in some cases. A common example occurs when the mixing process is linear and time-instantaneous.[3] For such problems, independent component analysis (ICA) can be used, see chapter 4. Many basic ICA methods require that the number of sensors cannot be smaller number of sources. When the number of sources is larger than the number sensors, we say that the problem is underdetermined. In some special cases, a solution to underdetermined source separation can be obtained using ICA algorithms, as long as the number of sensors is larger than two, $Q \geq 2$.

While ICA provides an elegant solution to multi-channel separation of linear instantaneous mixtures, it does not when the mixture model is in disagreement with the nature of the problem. For instance, in real-room acoustic mixtures, the source signals travel by multiple paths from the point of emission to the sensors, that is, there are multiple delays involved. As a consequence, a socalled convolutive mixture model is required to do any useful processing, complicating the algorithms significantly. The room impulse functions of the paths between the sources and the sensors are generally unknown and have to be estimated from data. Chapter 3 treats source separation in convolutive mixtures. Further complications of the mixing model in the form of non-linearities can occur if, for example, microphones are used as sensors, but this falls outside the scope of this text. Varying degrees of knowledge about the mixing process can be integrated into the model. In this thesis, the derived separation algorithms are mostly *blind*, indicating that the mixing process is unknown. However, the oft-used term, *blind source separation* seems to be somewhat of a misnomer, since a minimal set of assumptions always is implicitly assumed, typically linear mixing and the source dimensionality.

1.2 Applications

Within academia, a general interest in source separation has been demonstrated, as it provides researchers and scientists with a new tool to inspect phenomena of

[3]The instantaneous mixture at the j'th sensor can be described as $y_j(t) = \sum_i^P A_{ji} x_i(t)$, where t. As such, there are no dependencies across time in the observation model.

nature. For instance, it allows for previously unavailable views at seismic and cosmic data (Cardoso et al., 2002; Acernese et al., 2003). McKeown et al. (2003) reviews the application of ICA to brain images. Importantly, the algorithms used may apply to situations not predicted by their inventors, just as number theory is a foundation to the field of computer science.

In the shorter term, the research of source separation models and algorithms can be motivated from an applications point-of-view. Inspired by Mitianoudis (2004) and others, I provide a list of possible ways to exploit source separation algorithms in audio systems.

- In digital hearing aids, source separation may be used to extract the sounds of interest. This would constitute an improvement of today's beamforming methods, which merely perform directional filtering.[4] Taking advantage of communication between the devices at the left and right ears may boost the performance further of the source separation algorithm due to the increased distance between the sensors.

- In a number of cases, it is desirable to obtain transcriptions of speech. Sometimes, automatic speech recognition (ASR) can replace manual transcription, but in cross-talk situations and other noisy, adverse conditions the software may fail to provide useful results. It has been proposed that source separation could serve as a preprocessor to ASR, thus broadening the applicability of automatic transcription. A few examples of possible applications are: recordings of police interrogations, judicial proceedings, press conferences, multimedia archives.

Happy reading!

[4]Modern hearing aids are equipped with multiple microphones.

Chapter 2

Single-channel Separation

Generally, we cannot expect to be able to meaningfully map a single mixture signal into multiple separated channels. Rather it is a special feature of the source signals involved. For example, it has been demonstrated that a separating mapping can actually be performed on mixed speech (Roweis, 2001). This is not completely surprising, though, considering the fact that humans can separate speech from mono recordings, or at least, recognize the words (Cherry, 1953).

Paradoxically, the solution can be applied in a more general setting. For instance in audio scenarios, single-channel methods can be applied in all cases where a single microphone is already available in the hardware, such as cellphones and laptop computers. Multi-channel methods, on the other hand, would require versions of the appliances to be equipped with multiple microphones.

The chapter is organized as follows: first, single-channel separation is defined mathematically and issues of representation, preprocessing and postprocessing are addressed. Secondly, important methods of the relevant literature are mentioned and own contributions are placed in their proper context. Finally, a short discussion of the (subjective or objective) evaluation of algorithms follows.

In this thesis, only the linear version of the problem will be addressed, that is

$$y(t) = \sum_{i}^{P} a_i s_i(t) \tag{2.1}$$

where $y(t)$ is the mixture signal and $s_i(t)$ is the i'th source signal. In general,

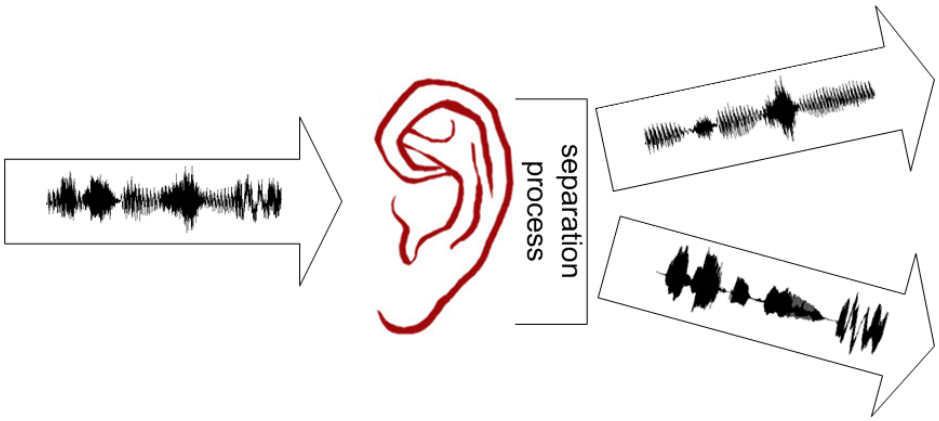

Figure 2.1: Single-channel separation is the art of mapping a single mixture of multiple sources into their components. Important inspiration can be taken from the human auditory system, which possesses a powerful ability to segregate and separate incoming sounds.

the gain coefficients, a_i, cannot be recovered and are assumed to be 1. This is due to a scaling ambiguity, which is inherent to the problem: from the point of view of $y(t)$ we can freely multiply a gain coefficient by a factor and divide the corresponding source signal with the same factor. In some situations, on the other hand, the powers of the sources can be assumed to have been acquired by some separate process and it is desirable to retain the a_i's in the model.

2.1 Preliminaries

The aim of machine learning methods (with which we are concerned) is to solve a given problem by adapting a general model to data. However, in practice the success often relies to a high degree on the preprocessing and postprocessing of the data, and to a lesser extend on the particular model applied. The search for suitable transformations of the problem can sometimes be described as 'linearization', suggesting that a difficult non-linear problem has been reduced to a simpler linear one which can be solved using our favorite, linear method. In fact, Michie et al. (1994) found that for 9 out of 22 different classification problems, linear discriminant analysis was among the best 5 out of 23 algorithms. The lack of robustness of complex non-linear models has to do with issues of generalization,

the models become overfitted to the training data. Motivated by such considerations, I will move on to describe feature representations of audio that has turned out to help achieve single-channel separation using machine learning methods. In reality, this indicates a compromise between knowledge-based and purist machine learning approaches.

In the context of single-channel separation of audio signals, it is common practice to use a time-frequency representation of the signal. Thus a the transformation, $\mathbf{Y} = \mathrm{TF}\{y(t)\}$, is performed as a preprocessing step. Often, $\mathbf{Y}$ is termed the 'spectrogram'. A common choice of calculating the TF, is the short-time Fourier transform (STFT), which efficiently computes amplitude and phase spectra on a time-frequency grid. It turns out that the phase spectrogram is irrelevant to many of the separating algorithms and may be imposed in an unaltered form to the outputted source estimates.[1] Hence, we define TF such that $\mathbf{Y}$ is a real-valued matrix with spectral vectors, $\mathbf{y}$, as columns. A common alternative option for computing TF is to employ a scale which has a high resolution at lower frequencies and a low resolution at higher frequencies, *e.g.*, that of a gammatone filterbank, or a mel scale. The mentioned TF mappings, which have turned out to be essential to obtain useful results, are clearly similar in spirit to the frequency analysis effectively carried out by the human auditory system (in the inner ear).[2] It is tempting to believe that this is not a coincidence: mimicking nature's way of sensing nature's signals may be near-optimal.

In order to qualify the usefulness of TF representations in audio processing, let us inspect the effect of the mapping on a sample. In figure 2.2, amplitude spectrograms of two audio are displayed along with their time-domain versions. The signals clearly become *sparse* in the TF domain, meaning that few of the TF cells are non-zero. This facilitates the separation of a mixture, because the energy of independent sources is unlikely to be overlapping. Further evidence is provided in figure 2.3, which shows the joint distribution of two speech sources, confirming the sparsity hypothesis. The chosen signals are quasi-periodic, meaning that most

[1] This is akin to spectral subtraction (Boll, 1979), a noise reduction technique for speech applications, which subtracts the estimated noise amplitude spectrum from the mixture amplitude spectrum. The 'noisy phase' carries over to the 'enhanced' signal.

[2] In a seminar session at the department, CASA pioneer DeLiang Wang reported that in his work on single-channel separation, the algorithms were relatively tolerant to the choice of TF.

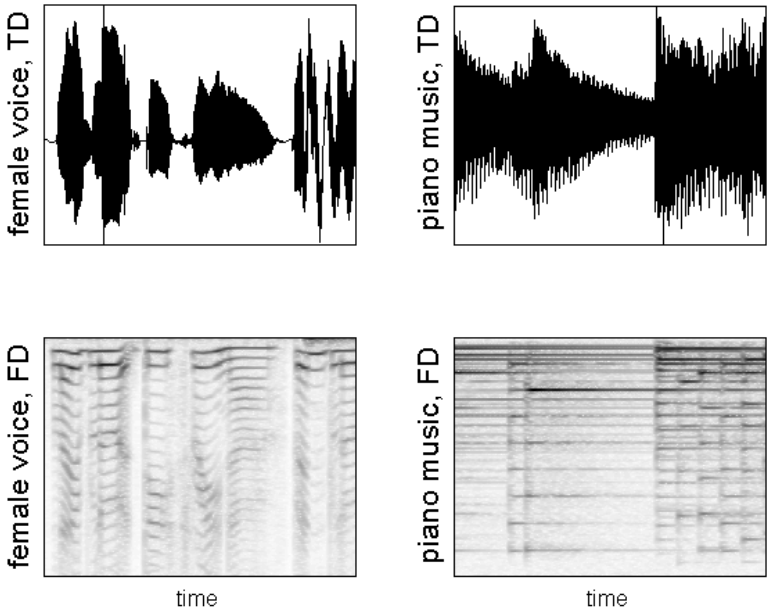

Figure 2.2: Time-domain (TD) and the corresponding TF representation (FD) of 2s excerpts from recordings of female speech and piano music (Beethoven). As a consequence of the mapping to the frequency domain, the signals become sparsely representable, that is, few elements are non-zero. The TF transformation were computed using the short-time Fourier transform.

segments of the signals are close to being periodic, a consequence of the speech production apparatus. As a result, the signals become sparse in the TF domain, *i.e.*, periodic signals are represented as 'combs'.

As a byproduct of the increased sparsity, linearity is approximately preserved in the transformed mixture,

$$\mathbf{y} \approx \sum_{i=1}^{P} a_i \mathbf{x}_i \tag{2.2}$$

where $\mathbf{x}_i$ is the transformed source signal. The time-index was dropped for ease of notation. Importantly, linearity enables a class of methods that rely on linear decompositions of $\mathbf{y}$, see section 2.6.

A further common practice in audio processing applications is to perform an

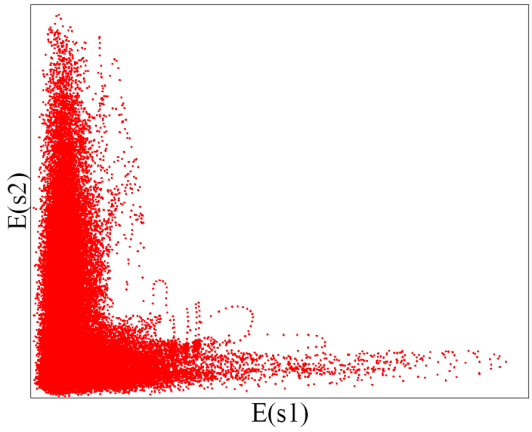

Figure 2.3: The energies at one frequency of two simultaneous speech signals in a TF representation, sampled at across time. It happens rarely that the sources are active at the same time. From Roweis (2003).

amplitude compression of $\mathbf{y}$, *e.g.*, by computing the squared cube root. This is biologically motivated by the fact that the human auditory system employs a similar compression, *e.g.*, as modelled by Stevens' power law (Stevens, 1957), and empirically motivated, see section 2.6.3.

We might consider the fixed resolution of the discussed TF transformations an unnecessary restriction. In fact, Gardner and Magnasco (2006) proposed that human audition uses a reassigned version of spectrogram, which adjusts the TF grid to a set of time-frequency points that is in closer accordance with the signal. In their framework, a pure sine wave is represented at its exact frequency rather than being smeared across a neighborhood of frequency bins. A delta-function (click) is similarly represented at its exact lag time. A major challenge in using the reassigned spectrogram for signal processing applications lies in adapting existing machine learning methods to handle the set representation (time-frequency-amplitude triplets). One possible solution is to quantize the reassigned spectrogram. This may, however, hamper the inversion to the time-domain.

2.1.1 Masking

The sparsification of signals via TF representation, which was described above, allows for an important class of solutions to single-channel separation that essen-

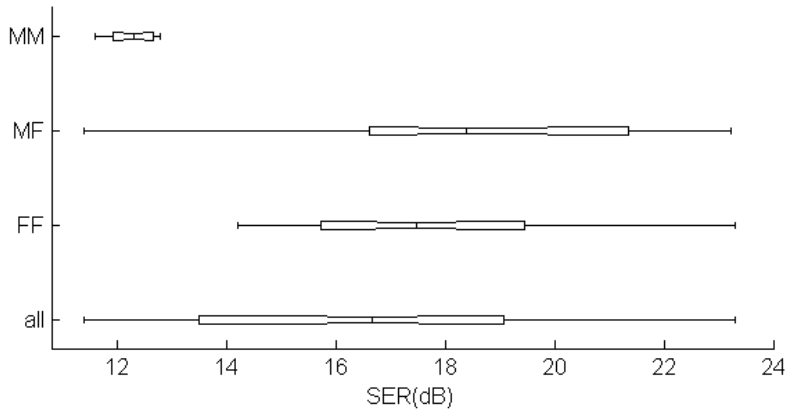

Figure 2.4: Single-channel separation of two speakers using ideal masks. Signal-to-error ratios (SER) in dB are reported for all combinations of 8 speakers from the GRID database. The SER figures were computed on a sample of 300s from each speaker. The ideal binary masks were constructed by performing a max-operation on the signal powers in the TF domain.

tially amounts to a (soft) classification of the TF cells. This is known as masking or refiltering (Wang and Brown, 1999; Roweis, 2001). For a given mixture, algorithm design effectively breaks down to (i) compute the TF representation, (ii) construct a mask, classifying all TF cells as belonging to either targets or interferers, and (iii) invert to the time-domain. The mask may be binary or 'soft', e.g., a probability mask.

I will proceed to estimate an upper bound on the performance of binary masking algorithms, which follows the scheme described above. To achieve this, a specific second step is assumed: The optimal mask is computed by simply assigning all energy of the mixture to the dominant source in each TF cell. This was done for 4 male and 4 female speakers from a speech database (Cooke et al., 2006). For all combinations of 2 speakers, a 0dB additive mixture of duration 300s was constructed. The mixtures were separated using ideal masks and the resultant signal-to-error ratios (SER) were computed. In figure 2.4, the figures are reported. The improvements as measured in SER are substantial, but more importantly, the masked speech sources *sound* almost completely separated. This can be explained by the hearing phenomenon of masking,[3] where one sound (A)

[3]Note that *masking* has two meanings: it is a separation method as well as an psychoacoustic

is inaudible due to the presence of a second sound (B). Frequency masking is one important case where the hearing threshold of (A) in a given frequency band is raised by the presence of (B) in the same band. Hence, the errors introduced by applying a binary mask to the mixture signal become largely inaudible due to the limitations of human hearing.

2.2 Filtering Methods

Noise-reduction techniques based on filtering have a history of being applied to, *e.g.*, audio applications. The objective is to estimate a target signal in noise. In this context, they can be viewed as a special case of single-channel separation algorithms, and the generative model of equation (2.1) reduces to,

$$y(t) = s(t) + n(t)$$

where $s(t)$ is the target signal and $n(t)$ is the interfering noise signal.[4]

Wiener (1949) proposed a method, which exactly minimizes the expected square error between inferred $\hat{s}(t)$ and $s(t)$, optimally infusing knowledge of the second-order-statistics of $s(t)$ and $n(t)$, which are further assumed stationary.[5] The Kalman filter (Kalman, 1960; Rauch et al., 1965) relies on nearly identical assumptions as formulated in a linear state-space model, but relaxes the stationarity requirement so that the solution is also optimal at the end-points of the time-series and across non-stationarities. From a Bayesian point of view, the Wiener/Kalman filter provides optimal inference when the signals involved are Gaussian and their distributions have been correctly specified.

Wiener and Kalman filtering are limited in their application due to the assumptions of stationarity and the availability of signal statistics. For instance,

phenomenon.

[4]While *filtering* is commonly associated with inference of $s(t)$ based exclusively on past and present observations of $y(t)$ and thus suited for real-time applications, *smoothing* includes future samples. *Prediction* uses only past samples. In this text, I use filtering as an umbrella term which includes smoothing and prediction.

[5]When applied in the time-domain, the Wiener filtering requires that the auto and cross-correlation functions of $n(t)$ and $s(t)$ are available. Sometimes it is beneficial to transfer to the Fourier domain. Then the variances at each frequency are assumed known.

in separation of multiple speakers, the second-order-statistics cannot be specified before-hand due to the fact that speech is non-stationary. However, if these can be provided through a parallel process, then Wiener filtering can play a role in single-channel speech separation (Benaroya et al., 2003). Speech can be regarded as stationary on the short term, and hence Wiener filtering can be applied to signal segments independently, provided that the required second-order-moments are available.

2.3 Incoherence

Cauwenberghs (1999) suggested to use phase incoherence in a separation algorithm, exploiting the effect of 'jitter' noise on the relative phase of the signals as well as that of amplitude modulation. The i'th source $s(t)$ is modelled as,

$$s_i(t) = B_i(t)p_i(t - \theta_i(t)) \tag{2.3}$$

where $p_i(t)$ is a periodic signal, $B_i(t)$ is the time-dependent amplitude, and $\theta_i(t)$ is the time-dependent phase. The key idea is to adapt sources that fulfill (2.3) for slowly varying random processes $B_i(t)$ and $\theta_i(t)$. Mutual independency is assumed of $B_i(t)$ and $\theta_i(t)$ across the sources, ideally restricting the estimated solution to the one sought.

Judging from the subsequent literature, the technique has not yet been widely applied, perhaps because it is limited to modulated periodic sources. Many real-life signals, such as audio sources, are non-stationary in ways that does not comply with (2.3). This does not exclude however, that phase (in)coherence as a grouping cue could be integrated in e.g. CASA methods, see section 2.7.

2.4 Factorial-Max Approximation

I will now return to methods that operate in the TF domain, where the sparsity of certain types of sources, notably speech, is exploited to the fullest extent. Roweis (2003) suggests a solution that extends on vector quantization, which in its basic form assigns each data point to the most similar prototype taken from a code-

book.[6] In order to be applied to single-channel separation of speech, as a first step, codebooks must be learned for each speaker. A socalled factorial vector quantizer is applied to decompose a given observed mixture vector into a sum of prototype vectors, one from each speaker codebook. However, this is a combinatorial problem which scales unfavorably with the codebook sizes, N_i. In fact, $\prod_i^P N_i$ likelihood evaluations must be performed for each mixture vector. This problem is further aggravated by the fact that we at (very) least we must require that $N_i \geq 100$ for all i in order to capture the variations of each speaker, *e.g.*, the pitch and phonemes. In order to alleviate the problem, the sparsity of the speech sources is formalized in the *max approximation*,

$$\mathbf{y} = \max\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_P\} \tag{2.4}$$

where $\max$ operates on the source vectors, $\mathbf{s}_i$, such that the output is the elementwise maximum. The max approximation combined with a white noise model allows for a substantial cut in the number of likelihood evaluations: elements of prototype vectors exceeding $\mathbf{y}$ incur an upper bound on the likelihood for all combinations including that particular prototype vector. Once the most likely component has been found for each source at each time, the sources are reconstructed using the masking technique.

Speech evidently has time structure (see figure 2.2), switching between fundamental states corresponding to atomic sound units, i.e. phonemes. In fact, each phoneme possesses a characteristic TF fingerprint. Hidden Markov models (HMM) employ state transition probabilities to quantify transitions between discrete states and associate an emission distribution to each of the states. HMM's have a long history of being employed to automatic speech recognition (Rabiner, 1989). Roweis (2001) suggested in his first paper on single-channel separation that each speech source should be modelled by a HMM. In analogy with the above discussion of the factorial vector quantizer, the resultant model of the observed mixture is a factorial HMM. Inference of the most probable state sequence is obtained via the Viterbi algorithm, but all combinations of source states need to be

[6] From a probabilistic point of view, mixture models such as Gaussian mixture models (GMM) can be regarded as a formalization of vector quantization. In fact, Roweis expresses his model in terms of a GMM.

considered at each time point. Hence, the number of likelihood evaluations is $\prod_i^P N_i$ in the naive implementation. However, the max approximation (2.4) can be used as described above, pruning the search tree dramatically. Roweis (2003) reports on the merits of the use of the more descriptive HMM model that 'in our experience the frame-independent MAXVQ model performs almost as well', an indication that dynamic models produce only modest improvements over time-instantaneous ones.

2.5 Inference in Factorial Models

One may argue that the factorial-max approximation is too removed from the reality of the signals. Kristjansson et al. (2004) did not compromise in the formulation of the generative model which assumes full additivity of the sources (assumed to follow a GMM) as well as a log-normal noise distribution. Instead, the source posterior was approximated by a Gaussian. Later, Kristjansson et al. (2006) extended the model to include HMM's describing the acoustical dynamics of speech as well as a dynamic language grammar model. This more sophisticated model helped achieve superior results in some cases. Evaluated on the GRID data set (Cooke et al., 2006), which contains speech sentences constructed from a limited vocabulary and grammar, the algorithm achieved a high level of separation, even in the hardest case of speaker separation where a speaker is (synthetically) mixed with itself.[7] Furthermore, the system performed better than humans in many cases.

Virtanen (2006b) also employs a factorial HMM, but suggests that the signals are represented by their mel-cepstral coefficients (MFCC). The computation of the MFCC in a time window consists of evaluating the power in the mel spectrum, taking the logarithm, performing a discrete cosine transform and retaining (the lower) part of the coefficients. MFCCs can be regarded as a compact representation of the spectral envelope and are often used directly in automatic speech recognition. Importantly, the MFCCs are insensitive to pitch variation. However, they do not preserve the linearity such as the high-resolution TF mappings discussed up until this point. Thus, the factorial-max approximation does not apply,

[7] Applications of same-speaker separation are arguably limited, but results may extend to cases where the speakers are acoustically similar.

and instead, the source MFCCs are inferred by imposing a log-normal approximation on their sum. Furthermore, a scheme is suggested to synthesize the source signals from their MFCCs.

2.6 Sparse Factorization

The vector quantization approaches described above are limited in the sense that the mixture vector at any time is modelled as a sum of prototype vectors, one for each source.[8] This restriction can be relaxed by employing a factorization model, that is, the contribution of each source is a weighted sum of prototypes. The i'th source, $\mathbf{s}_i$, is decomposed in terms of a dictionary (or, codebook) $\mathbf{d}_{ij}$ and its encodings c_{ij},

$$\mathbf{s}_i = \sum_{j=1}^{N_i} \mathbf{d}_{ij} c_{ij} = \mathbf{D}_i \mathbf{c}_i \tag{2.5}$$

where the dictionary matrix $\mathbf{D}_i$ holds the $\mathbf{d}_{ij}$ in its columns, and $\mathbf{c}_i$ is defined accordingly. The combination of the models (2.2) and (2.5) results in,

$$\mathbf{y} = \sum_{i=1}^{P} \mathbf{D}_i \mathbf{c}_i = \mathbf{D}\mathbf{c} \tag{2.6}$$

The number of dictionary elements, $\sum_i N_i$ is allowed to be larger than the dimensionality of $\mathbf{y}$, meaning that $\mathbf{D}$ is potentially overcomplete, *i.e.*, many possible decompositions exist. This has been shown to result in more natural and compact representations (Olshausen and Field, 1996).

In order to apply the factorization (2.6) to the problem of signal separation, two decoupled steps must be completed: a set of dictionaries, $\mathbf{D}_i$, is learned from a training set of unmixed $\mathbf{x}_i$ as a first step. Subsequently, the joint encoding, $\mathbf{c}$, is computed on the basis of the concatenated source dictionaries, $\mathbf{D}$. Finally, the sources are re-synthesized according to 2.5. The method assumes that the dictionaries of the sources in the mixture are sufficiently different. When this is

[8] The narrative flow is inspired by the one used in (Asari et al., 2007)

not the case, they do no become separated in the encoding.

Different matrix factorization methods can be conceived based on various a priori assumptions of the dictionaries and encodings. Since computing $\mathbf{c}$ (given $\mathbf{D}$) from 2.6 is generally ill-posed, the model should at least impose sufficient constraints for the inversion to produce a well-defined solution. Jang and Lee (2003) applied independent component analysis (ICA) in the time-domain to learn the dictionaries from unmixed audio data and later employed them to a sparse decomposition of the mixture signal, achieving a level of separation. Similarly Benaroya et al. (2003) used sparse non-negative matrix factorization (NMF) to learn dictionaries from isolated recordings of musical instruments and compute a decomposition. Smaragdis (2004, 2007) also uses NMF, but further extends the model to a convolutive version in order to capture atoms that have a time-structure.

Some methods combine the learning of the dictionaries and the encoding into a single stage. Casey and Westner (2000) projects the mixture spectrogram to a subspace and then performs ICA. The ICs are projected back into the original space and clustered, forming source estimates. The algorithm provides an alternative in cases where samples of the isolated sources are unavailable, but it should be expected that the method would require a larger sample to learn the optimal basis functions.

Alternatively, it has been shown that source signals from identical distributions can be separated provided that information about the signal path is available (Asari et al., 2006). In an audio context, this is essentially an extension of equation 2.2 to a *convolutive* model in the time-domain. In the TF domain this translates to a multiplicative modification of the dictionary of the i'th source,

$$\widetilde{\mathbf{d}}_{ij} = \mathbf{h}_i \bullet \mathbf{d}_j \tag{2.7}$$

where $\mathbf{h}_i$ is the frequency response of the path between the i'th source and the microphone and $\bullet$ indicates elementwise multiplication. The modified dictionaries, $\widetilde{\mathbf{d}}_{ij}$, provide additional contrast for 'similar' sources, but require knowledge of the signals paths.

2.6.1 Contribution I

Pearlmutter and Olsson (2006) explore sparse decompositions for single-channel speech separation. The observation model is identical to equation (2.6), and the assumed prior distribution of the coefficients is i.i.d. Laplacian. This model formulation leads to an L1 norm optimization problem which can be solved using linear programming (LP). In fact, LP is used to (i) learn the dictionaries, and (ii) compute the sparse decomposition required in (2.6) for the separation of the sources. The first is achieved through a stochastic-gradient (Robbins and Monro, 1951) optimization of the (L1) sparsity of the decomposition. The second amounts to a version of basis pursuit (Chen et al., 1998). The paper has been incorporated into a book chapter on sparse single-channel separation (Asari et al., 2007).

2.6.2 Contribution II

Essentially attacking the same problem as above, we (Schmidt and Olsson, 2006) exploit the fact that all quantities involved in the TF domain decompositions are non-negative. We use a sparse version of non-negative matrix factorization (Eggert and Körner, 2004) to learn the dictionaries as well as to compute a separating decomposition. This implies a Gaussian model for the error in equation (2.6) and a one-sided exponential prior distribution for the coefficients. Virtanen (2006a) mentioned our article.

2.6.3 Contribution III

Generative models are often used to motivate particular applications of ICA, NMF or sparse decompositions, *e.g.*, we may say that the coefficients are mutually independent and follow a long-tailed distribution. In reality, these models are often mismatched to the data. For instance, linearity might not hold. Sometimes we do not get independent components from ICA but rather a set of inter-dependant features. We (Schmidt and Olsson, 2007) suggest to perform linear regression on non-linear features (*e.g.*, the NMF used in Schmidt and Olsson, 2006), achieving a performance boost over naive re-synthesization from the features.

2.7 CASA methods

The model-based approaches described so far attempt to learn structure from data and apply the models to the inversion from the mixture to the sources. Alternatively, separation algorithms can be designed to emulate the sound segregation processes of the human auditory system, that is, perform computational auditory scene analysis (CASA, Bregman, 1990). Working in the TF domain, Hu and Wang (2003) proposes a method, which can extract a speech signal from a mixture. In a number of stages, the TF cells are grouped according to cues such as temporal continuity, correlation across channels and periodicity. By visual inspection of, *e.g.*, figure 2.2, it is clear that speech patterns ('harmonic stacks') lend themselves to these affinity measures. The higher frequencies are treated separately, assigning them to the grouping established in the lower frequencies based on amplitude modulation patterns. The method works better for intrusions other than speech (and similar signals), due to the fact that the employed segregation mechanisms are specifically designed to send the speech parts to the foreground stream.

Bach and Jordan (2005) perform clustering of the TF elements based on parameterized distance measures inspired by CASA. The parameters of the distance measures are adapted to a training set.

2.8 Algorithm Evaluation & Comparison

Many of the described algorithms are developed from a machine learning outset, where the goal is to maximize the signal-to-error ratio (SER) on the test set: the higher the better.

However, in audio applications, the evaluation should take into account how the output of the algorithm would *sound*. Thus, a source separation algorithm should be evaluated according to the degree to which the sounds are perceived as separated. A related issue is audio coding such as MP3,[9] where an increased SER is acceptable, so long as the deteriorations are inaudible to a human listener.

[9] Short for MPEG-1 Audio Layer 3

Conversely, serious artifacts in the processed audio caused by some algorithms may result in relatively small decline in SER.

Ideally, the output of all the mentioned algorithms for single-channel separation of speech should be exposed to human subjective evaluation. In the case of speech, the second best solution may be to expose the algorithms to a standard automatic speech recognizer (ASR). This was done in the 2007 Speech Separation Challenge.[10] However, this approach has its own inherent weakness in that the ASR may exhibit an undesired pattern of sensibilities. Ellis (2004) discusses the evaluation of speech separation algorithms.

One might speculate that a purist Bayesian machine learner might dislike the idea of using different cost-functions for learning parameters and for evaluating those. A more fundamentally sound approach would consist in optimizing a distance measure which is founded on the proper psychoacoustic principles.

[10] See `http://www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.htm`.

Chapter 3

Multi-channel Separation

Multiple sensors are exploited in naval surveillance, where hydrophone arrays are used to map the positions of vessels. In electroencephalography (EEG), electrodes are placed on the scalp to monitor brain activity. Similarly, modern hearing aids are equipped with multiple microphones. It is common to these examples that the intensity interfering signals is significant in relation to the target signals. Multiple sensors are used to amplify signals originating from a given direction in space and to suppress the signals from other directions, thus increasing the target-to-interferer ratio. In its basic form, this is known as beamforming, a term which usually refers to linear array processing and can be regarded as a spatial generalization of classical filtering techniques (Krim and Viberg, 1996). More generally, signal separation algorithms, linear as well as non-linear, may benefit from the added discrimination power provided by multiple sensors and this is indeed the topic of the chapter.

The content is organized as follows: the convolutive model for multi-channel mixtures is defined in in section 3.1. The major part of the coverage focuses on methods that are based on second-order statistics, or, Gaussian signal assumptions (section 3.2). Other methods, *e.g.*, those based on higher-order statistics and non-Gaussian distributions, are reviewed briefly in section 3.3. Comments on published work co-authored by me are situated in the vicinity of the their relatives in the literature.

3.1 Scope and Problem Formulation

In the context of separation of audio signals, multiple microphones have been employed with some level of success. Weinstein et al. (1993); Yellin and Weinstein (1996) provide the earliest evidence that speech signals could be separated from their mixtures, which were recorded in a real room. Interest in the field has since surged, so much that Pedersen et al. (2007) can cite 299 articles on the subject. The count is much higher if the more general problem of multi-channel separation is considered: At the 2006 conference on Independent Component Analysis (ICA) and Blind Source Separation in Charleston, 120 papers were presented.[1] This is the sixth meeting on the topic since 1999. The major part of the research is concerned with blind separation of instantaneous linear mixtures, that is, given the observation model $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, estimate $\mathbf{A}$ and infer the sources $\mathbf{s}(t)$. Under assumptions of independency and non-Gaussian sources, this problem can sometimes be solved using ICA, see chapter 4.

The coverage here, on the other hand, is exclusively devoted to the set of problems that are best described by a convolutive model,

$$\mathbf{y}(t) = \sum_{\tau=0}^{L-1} \mathbf{A}(\tau)\mathbf{s}(t-\tau) + \mathbf{v}(t) \tag{3.1}$$

where the observed $\mathbf{y}(t)$ is a vector of mixture signals at time t, $\mathbf{s}(t)$ and $\mathbf{v}(t)$ are the source and noise vectors, respectively. The mapping is governed by $\mathbf{A}(\tau)$, which is a set of mixing matrices at L different lags. Assuming that the sources, $\mathbf{s}(t)$, are mutually, statistically independent and that the channel, $\mathbf{A}(\tau)$, is unknown, the overall goal is to estimate $\mathbf{A}$ and infer $\mathbf{s}(t)$.

The convolutive model arises when the mixture is not instantaneous, that is, when the sources mix into the sensors as filtered versions. One instance of this arises when there are different time-delays between a given source and the sensors. This naturally occurs in acoustics scenarios, *e.g.*rooms, where the sounds travel different distances between the sources and the sensors, and, additionally, multiple echoes of an emitted sound are observed at a sensor (see figure 3.1). In acoustic

[1]The conference web site is located at `http://www.cnel.ufl.edu/ica2006/papers_accepted.php`

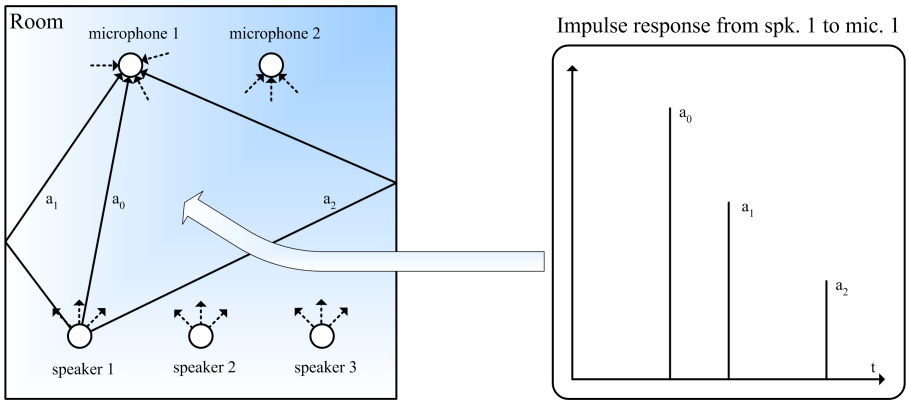

Figure 3.1: The convolutive mixing model exemplified: the sounds are reflected by the walls of the room and arrive at the microphones with various delays and attenuations. The corresponding observation model is a convolution sum of the source signals and the impulse responses.

mixtures, we can thus regard $(\mathbf{A})_{ij}(\tau)$ as describing the room impulse response between source j and sensor i. In general, the model cannot be inverted, and the sources cannot be retrieved, but a solution exists in many special cases, which are described in the following sections.

Nothing entirely general can be said about the identifiability of the sources and the channel, since it naturally depends on the assumptions included in the separation algorithm. However for the set of methods that assume little, *e.g.*, that the sources are independent or uncorrelated, the source signals, $\mathbf{s}(t)$, can be determined only up to an arbitrary filtering. This is because filtered versions of the room impulse functions in $(\mathbf{A})_{ij}(\tau)$ may be cancelled by applying the inverse filter to $(\mathbf{s})_j(t)$. However, if the source separation algorithms have been informed of, *e.g.*, the scale or the coloring of $\mathbf{s}(t)$, the ambiguity is reduced accordingly. Sometimes the arbitrary filtering of the inferred sources is undesirable, and we may choose to project back to the sensor space, in which case the ambiguities in $(\mathbf{A})_{ij}(\tau)$ and $(\mathbf{s})_j(t)$ cancel out. Practically speaking, this means that we infer the audio sources as they sound at the microphones.

Furthermore, the source index may be permuted arbitrarily, in that the model is invariant to a permutation of the elements of $\mathbf{s}(t)$ and the columns of $\mathbf{A}(\tau)$. In the case of equal number of sources and sensors ($Q = P$), we can only hope to

estimate $\mathbf{P}\mathbf{s}(t)$ and $\mathbf{A}(\tau)\mathbf{P}^{-1}$, where $\mathbf{P}$ is a permutation matrix.

An important simplification occurs when the convolutive mixing model (3.1) reduces to a pure attenuate-and-delay model, where only a single filter tap is non-zero. In this case, the i,j'th element of $\mathbf{A}(\tau)$ is redefined as

$$\left(\widetilde{\mathbf{A}}\right)_{ij}(\tau) = \delta(\tau - \Delta_{ij}) \tag{3.2}$$

where $\delta(\tau)$ is the Kronecker delta function and Δ_{ij} is the delay involved between the j'th source and the i'th sensor. Acoustic mixing in an anechoic room is appropriately represented by (3.2).

3.1.1 Frequency Domain Formulation

Many algorithms work in the (Fourier) frequency domain, where multiplication approximately replaces convolution. Therefore, I redefine (3.1) by applying the discrete Fourier transform (DFT) to windowed frames of $\mathbf{y}(t)$, obtaining,

$$\mathbf{y}_k^{(n)} = \mathbf{A}_k \mathbf{s}_k^{(n)} + \mathbf{e}_k^{(n)} \tag{3.3}$$

where $\mathbf{y}_k^{(n)}$, $\mathbf{s}_k^{(n)}$ and $\mathbf{A}_k$ are the frequency domain versions of the corresponding time-domain signals at discrete frequencies k. The window (time) index is n. There is a residual term, $e_k^{(n)}$, which is partly due to additive noise, $\mathbf{v}(t)$, and partly due to the fact that equation 3.1 is a linear convolution rather than a circular one. When the window length is much larger than L, the latter mismatch vanishes, that is $\langle \frac{|\mathbf{e}_k|}{|\mathbf{x}_k|} \rangle \rightarrow 0$. The notation used indicates that the channel, $\mathbf{A}_k$, is assumed constant on the the time-scale of the estimation, which may sometimes be a rather strict constraint, *e.g.*, excluding a cocktail party situation with overly mobile participants.

3.1.2 Frequency Permutation Problem

The transformation to the frequency domain is particularly useful, because it allows efficient ICA methods to be applied independently to each bin, k, in equation 3.3. However, there is a serious challenge associated with following such an

approach, namely that the permutation problem (described above) also becomes decoupled across frequencies. This has the consequence that the inversion to the time-domain has been made difficult unless the permutation can be harmonized, so that it is the same for all bins. Assumptions regarding the channel and the sources can be exploited for this purpose. Consider for example a pure delay-and-attenuate mixing system (3.2), which can be regarded as modelling an anechoic room. Then the estimated $\hat{\mathbf{A}}(\tau)$ should be sought permutation-corrected so that the amplitude is constant across frequency and the phase is linear in frequency.

Alternatively, the frequency permutation problem can be fixed by using the structure in the sources. One possibility is to optimize the correcting permutation so that it maximizes the correlation of the amplitudes across frequencies. In fact, Anemüller and Kollmeier (2000) turned this criterion into a full separation algorithm.

3.2 Decorrelation

In signal processing, it is a common theme to base a solution on the second-order statistics of the signals. Ignoring the means, which can be pre-subtracted and post-added, this means that the relevant information is contained in the auto and cross-correlation functions. In the context of multi-channel separation, this translates to ensuring that the cross-correlation between the sources is zero at all lags. The time-lagged covariance of the source estimate $\hat{\mathbf{s}}(t)$ is defined

$$\mathbf{\Lambda}(\tau) = \left\langle \hat{\mathbf{s}}(t)\hat{\mathbf{s}}^{\top}(t-\tau) \right\rangle \tag{3.4}$$

where τ is the lag time. The goal is to diagonalize $\mathbf{\Lambda}(\tau)$. Molgedey and Schuster (1994) showed that for instantaneous mixtures (those that are constrained to $L = 1$ in equation 3.1) diagonalization in fact retrieves the actual sources, except for a scaling and permutation uncertainty. In fact, they showed that $\mathbf{\Lambda}(\tau)$ is only required to be diagonal at $\tau = 0$ and additionally at a lag different from zero $\tau = \tau_0$. The solution to $\mathbf{A}$ is obtained by solving an eigenvalue problem.[2] It is a condition that the ratio between the auto-correlation coefficients at these

[2]It is assumed that $\mathbf{A}$ is invertible.

lags is different across sources in order for the problem to be solvable using this technique. Parra and Sajda (2003) generalized the eigenvalue solution to other statistics than lagged covariance matrices, providing a quick-and-dirty method in many instances.

In the case of the full convolutive model (3.1), the decorrelation of stationary sources does not achieve the identification of the mixing system or the inference of the sources as noted by, *e.g.*, Gerven and Compernolle (1995). This can be realized by considering the decorrelation criterion (3.4) in the frequency domain. The auto/cross power spectra of $\mathbf{x}_t^{(n)}$, $\mathbf{C}_k^{(n)}$, depend on the spectra of $\mathbf{s}_t^{(n)}$ as follows,

$$\mathbf{C}_k = \mathbf{A}_k \mathbf{D}_k \mathbf{A}_k^H + \mathbf{E}_k \tag{3.5}$$

where $\mathbf{D}_k^{(n)}$ is a diagonal matrix with the powers of the sources as elements. The power spectrum residual, $\mathbf{E}_k$ vanishes when $\mathbf{e}_k$ is small. Now it can be seen that the channel and the source spectra are ill-determined because $\{\mathbf{A}_k, \mathbf{D}_k\}$ and $\left\{\mathbf{A}_k \mathbf{U} \mathbf{D}_k^{\frac{1}{2}}, \mathbf{I}\right\}$ are solutions that produce identical statistics, $\mathbf{\Lambda}(\tau)$ and hence indistinguishable. The orthogonal matrix, $\mathbf{U}$, obeys to $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$. Hence, additional discriminative properties of the sources need to be present in order to overcome this limitation.

In order to identify the model, Weinstein et al. (1993) suggested to take advantage of a fairly common quality of real-world signals, namely that their statistics vary in time. For example, speech signals can be considered non-stationary if measured across windows that are sufficiently short (but still long enough to obtain a reliable estimate). Thus, we extend (3.5) to account for the non-stationarity,

$$\mathbf{C}_k^{(m)} \approx \mathbf{A}_k \mathbf{D}_k^{(m)} \mathbf{A}_k^H \tag{3.6}$$

where m is the window index not the be confused with the index in (3.3). The key point is that, if different auto/cross power spectra are measured at multiple times (with $\mathbf{A}_k$ fixed), then the the number of constraints increase at a higher rate than the number of unknowns. Parra and Spence (2000) turned (3.6) into a practical algorithm employing gradient descent as the vehicle of optimization. The problem of different permutations across frequency was approached by constraining the

filter length, L, to be sufficiently smaller than the window length of the DFT, effectively ensuring smooth frequency responses.

Rahbar and Reilly (2005); Olsson and Hansen (2006a) note that the non-stationary observation model (3.6) fits in the framework of multi-way analysis (Smilde et al., 2004). This can be seen by comparing to the symmetric version of the parallel factor (PARAFAC) model which is defined $x_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} a_{kf}$, where a_{if} and b_{jf} are the loading matrices and F is the number of factors. The loading matrices have been shown to be identifiable for quite a high number of factors, lower bounded by a theorem by Kruskal (1977). The treatment of (3.6) may still be to gain further from the body of analysis and algorithm accumulated in the field of multi-way analysis.

3.2.1 Contributions IV-VI

Cost-functions which depend on second-order-statistics only often result from placing Gaussian assumptions on the variables of a linear generative model. In my work on time-domain algorithms, I indeed assumed Gaussianity and was able to derive maximum posterior (MAP) inference for the sources and maximum-likelihood estimators for the parameters. A linear state-space model which allows time-varying parameters was employed, including an autoregressive (AR) process with Gaussian innovation noise as a source model.[3] Olsson and Hansen (2004b) applied maximum-likelihood learning to the parameters of the model using an expectation-maximization (EM) algorithm to do so (Dempster et al., 1977). On the E-step, the sources are inferred using the Kalman smoother. The parameters are re-estimated on the M-step. In order to reach convergence, the E and M steps were invoked alternatingly. We successfully separated speech signals that were mixed in a convolutive model and showed that the method is resilient to additive Gaussian noise. As an integral part of the Kalman filter implementation, the likelihood of the model parameters given the observed data is computed in the process of inferring the sources. This can be used in a model control framework, where the objective is to estimate the number of active sources in each time-window. For this purpose, Olsson and Hansen (2004a) employed the Bayesian Information

[3] See the papers for details.

Criterion (BIC, Schwartz, 1978), which is an approximation of the Bayes factor/marginal likelihood of the model. The main computational component in BIC is the likelihood computation.

An effort was made to tailor the algorithm to a specific domain, namely the separation of speech signals. For that purpose, a native part of linear-state space models, known as the *control* signal, can be used to shift the mean of the innovation noise process that drives the sources. Olsson and Hansen (2005) used a parameterized speech model as a control signal, effectively attracting the solution to be in agreement with the speech model. We used the model of McAulay and Quateri (1986), who coded fragments of speech signals in terms of a sum of a period signal and colored noise. As a necessary addition to the algorithm, the time-varying fundamental frequencies and harmonic amplitudes and phases are estimated.

Zhang et al. (2006) extended our algorithm to account for a non-linear distortion of the observed mixtures and showed that the new method performs better than ours on synthetic data. Särelä (2004); Chiappa and Barber (2005); Pedersen et al. (2007) referred to our work on this topic.

3.2.2 Contribution VII

Having formulated our favorite generative model of data, it is often a major obstacle to choose the parameters of that model. In this case and in many other cases, there are a number of unobserved sources or missing data which influence the model. This precludes direct maximum-likelihood (ML) learning, as the complete likelihood function depends on data which are unavailable. Rather, the *marginal* likelihood should be optimized, requiring the formulation of a prior probability distribution for the sources. However, the resulting marginalization integral may not be easily optimized with respect to the parameters. The EM algorithm is an iterative approach to obtaining the ML estimate, both in terms of simplicity of analysis and ease of implementation.

Slow convergence is a major caveat which is associated with the EM algorithm but also with, *e.g.*, steepest gradient descent. We (Olsson et al., 2007) discuss the possibility of extracting the gradient information from the EM algorithm and

feeding it to an off-the-shelf, state-of-the-art Newton-type optimizer. The result is a sizable speedup for three different problems. Pontoppidan (2006) noted our work.

3.2.3 Contribution VIII

Beside summarizing the works mentioned above, we (Olsson and Hansen, 2006b) introduce stochastic gradient (SG) learning for the parameters of the state-space models. It is well-known that SG can reduce significantly the computation time to reach convergence when the number of data-points is large. For the state-space model whose parameters vary in time, the number of parameters is proportional to the length of the supplied audio sample. Thus, the gradient method and EM algorithm are impractical for large data volumes, whereas SG is well suited.

Furthermore, the potential benefit of incorporating the detailed speech model is documented, namely that the learning of the parameters may converge faster than a (blinder) baseline method. Some caution should be given to the fact that the proposed algorithm suffers from local minima of the cost function and a high computational intensity.

3.3 Other methods

The previous section dealt with methods that are based on second-order-statistics, which in many cases is similar to placing Gaussian assumptions on the source signals and deriving the according estimators. Naturally, other distributions can pose as source models. In fact, evidence from problems that are best described by linear instantaneous mixtures, strongly suggests that non-Gaussianity helps identify the mixing matrix and thus facilitates the separation of the sources (see the chapter on independent component analysis (ICA), 4). In this connection, it is a fortunate fact that many real-life signals are non-Gaussian, *e.g.*, speech follows a long-tailed, sparse distribution (see figure 2.3 in chapter 2).

A significant number of authors describe algorithms which address the generalization of ICA to convolutive ICA (CICA). Already in 1999 the number is considerable, Torkkola (1999) cites 115 works in his paper ‘Blind separation for

audio signals - are we there yet?', predominantly CICA references.

Thi and Jutten (1995) generalized the decorrelation approach to include the minimization of the magnitude of higher-order moments. The mixing matrix can be identified up to the usual permutation and filtering ambiguities, also in the case of stationary sources, which could not be treated by decorrelation.

A number of authors, *e.g.*, Pearlmutter and Parra (1997) and Moulines et al. (1997), formulated the problem in turns of a generative model, specifying densities for the sources. Subsequently, the parameters are estimated by likelihood function optimization. Attias and Schreiner (1998) derived algorithms from probabilistic principles in the time-domain as well as in the frequency-domain. They noted, as was pointed out in the previous section, that the frequency permutation problem could be made less severe by constraining the filter length, L, to be much smaller than the window length.

A separate issue is the functional form of the source inference, which is sometimes subjected to a deliberate design choice. In hardware applications, for instance, it may be beneficial to obtain a mathematical function that fits into a multiply-and-add framework. The noise-free version of (3.1) has a recursive solution,

$$\hat{\mathbf{s}}(t) = \mathbf{A}(\tau)\left(\mathbf{y}(t) - \sum_{\tau=1}^{L-1}\mathbf{A}(\tau)\hat{\mathbf{s}}(t-\tau)\right) \tag{3.7}$$

that is, if the mixing process is invertible. However, linear systems of this type (infinite impulse response, IIR) are known to suffer from instability in some cases. Lee et al. (1997) and Dyrholm et al. (2007) discuss the advantages and disadvantages of the IIR solution as contrasted with a finite impulse response (FIR) separator,

$$\hat{\mathbf{s}}(t) = \sum_{\tau=0}^{L'-1}\mathbf{W}(\tau)\mathbf{y}(t-\tau) \tag{3.8}$$

It should be noted that optimal inference of the sources in (3.1) is a generally non-linear endeavor, the functional form depending on the assumptions made.

An important and appealingly simple approach to convolutive ICA is to apply

ICA independently to each frequency bin in (3.3). Among the authors, which have experimented with ICA in the frequency domain, are Murata et al. (2001), who applied the Molgedey and Schuster (1994) method to each frequency bin. The permutation problem was attacked by maximizing amplitude correlation across frequency as mentioned in section 3.1.1.

3.3.1 Masking Methods

In chapter 2, I discussed turning a particular property of speech to our advantage, namely that it is sparse in a time-frequency (TF) representation. As a result, the max-approximation applies, and each TF cell is likely to contain energy deriving from, at most, a single source.

A number of multi-channel separation methods exploit this fact by treating the problem as that of assigning each cell to a source. In a blind setup, where the channel is unknown, this amounts to performing clustering of a feature mapping on the mixture TF representations. Yilmaz and Rickard (2004) treat blind separation of many speech sources which have been mixed into two channels. The mixture model is attenuate-and-delay (3.2) as opposed to full convolutive mixing, corresponding to anechoic room mixing. A two-dimensional feature representation, based on the ratio of amplitudes and the phase differences between the channels, is subsequently used to group the sources (see figure 3.2). Similar features have been suggested from a CASA point-of-view, where interaural time/intensity differences (ITD/IID) are the preferred terms corresponding to psychoacoustic quantities (Roman et al., 2004). It is important to note that the ITD is ambiguous at higher frequencies, as the wavelength decreases below the difference in travelled distance to the ears.

3.3.2 Contribution IX

Following in the footstep of, *e.g.*, Bofill and Zibulevsky (2001), Araki et al. (2003) and Yilmaz and Rickard (2004), we (Olsson and Hansen, 2006a) attack the problem of separating more sources than sensors in convolutive mixtures. The algorithm, which works in the frequency domain, exploits the non-stationarity of speech and applies k-means clustering to IID/ITD-like features at each frequency

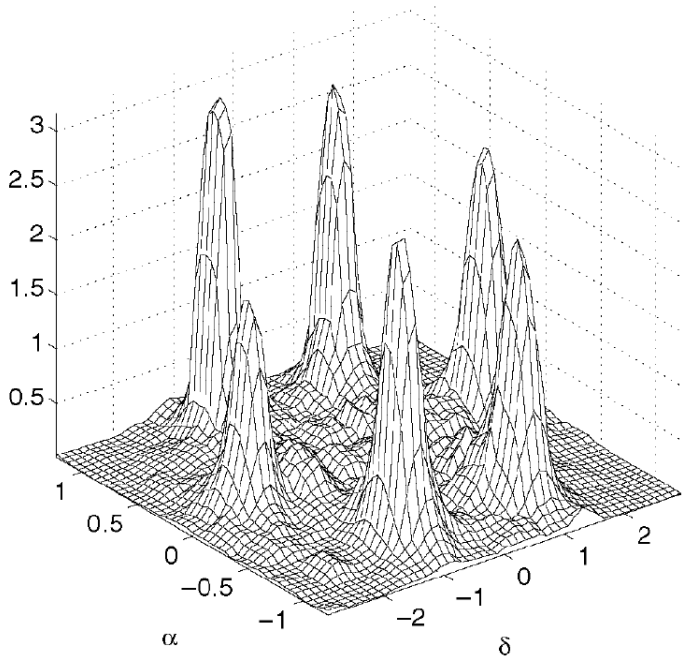

Figure 3.2: The empirical distribution of amplitude, α, and delay variables, δ, for a attenuate-and-delay mixture of 6 speech sources. The α and δ correspond to interaural intensity and time differences, respectively (IID/ITD). The peaks of the distribution correspond to the sources and can be used to construct a TF mask, which assigns the energy to 6 different channel, allowing for the separation of the sources. From Yilmaz and Rickard (2004).

separately. As a result, a permuted version of the channel, $\mathbf{A}_k$, is estimated along with the power spectra of the sources, $\mathbf{D}_k^{(m)}$. The permutation is corrected by greedily maximizing the amplitude correlation within a source. Subsequently, the sources are inferred by Wiener filtering, benefitting from having estimated the relevant statistics. In controlled conditions, the results are excellent. However, in a real reverberant room, the sparsity of the speech at the microphone may be too low to achieve overcomplete separation (more sources than sensors).

Chapter 4

Independent Component Analysis

Whereas source separation is a designation assigned to a class of problems, independent component analysis (ICA) is more often used to refer to a more restricted set of *methods*. For instance, Comon (1994) states that 'the independent component analysis (ICA) of a random vector consists of searching for a linear transformation that minimizes the statistical dependence between its components'. Research in ICA and related topics have surged and there are now multiple textbooks on the subject, *e.g.*the one by Hyvärinen et al. (2001).

In the following, I will briefly describe ICA as it may be defined from a generative model point-of-view. By this is meant that parameterized probability density functions are assumed for the involved stochastic variables, from which we can draw samples. When a generative model has been formulated, the derivation of statistical inference such as maximum likelihood (ML) or maximum posterior (MAP) is often mechanical (MacKay, 1996; Højen-Sørensen et al., 2002). The assumptions are as follows

1. The observable is a linear mixture of the source signals,

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{v} \tag{4.1}$$

   where $\mathbf{y}$ is the mixture vector, $\mathbf{A}$ is the mixing matrix, $\mathbf{s}$ is the source vector and $\mathbf{v}$ is additive noise.

2. The sources are mutually independent, that is, the prior probability density

function factorizes, $\mathrm{p}(\mathbf{s}) = \prod_i^P \mathrm{p}(s_i)$, where s_i are the individual sources.

3. The sources are distributed according to non-Gaussian probability density functions. The noise, $\mathbf{v}$, may be zero, or something else.[1]

Having stated the assumptions, ICA can simply be defined as: given a sample $\{\mathbf{y}^n\}$, infer $\{\mathbf{s}^n\}$. In the case of zero-noise conditions and an equal number of sources and sensors ($P = Q$), and invertible $\mathbf{A}$, ICA simplifies to two steps. The first step is to estimate $\mathbf{A}$, *e.g.*in ML fashion, where the likelihood function, $\mathrm{p}(\mathbf{y}^n|\mathbf{A})$, is optimized. The second step is to map back to the source space, $\mathbf{s} = \mathbf{A}^{-1}\mathbf{y}$. (MacKay, 1996) derives efficient update rules for the inverse of $\mathbf{A}$ that are based on ML learning.[2]

It is apparent that the scale of the s_i cannot be estimated from data alone, just as the ordering in the reconstructed source vector is undeterminable. These are known as the scaling and permutation ambiguities.

A much broader definition of ICA is sometimes given rather than the narrow linear and instantaneous[3] definition stated above. Alternatively, taking the acronym 'ICA' more literally we could define it simply as: invert a general mapping of the sources to the mixtures. Obviously, this is in general impossible, but specialized solutions have been proposed, *e.g.*, for convolutive mixtures (Pedersen et al. (2007) provides a comprehensive review).

4.1 Why does it work?

While this question is addressed in detail by Hyvärinen et al. (2001), I will give a brief, informal summary of the key points. First of all, ICA can be viewed as a generalization of principal component analysis (PCA) where data is linearly transformed to the subspace that retains the largest variance. Roweis and Ghahramani (1999) describes PCA in terms of a generative model, where the assumptions are

[1] In fact, it is permissable that at most 1 source is Gaussian.

[2] Originally, Bell and Sejnowski (1995) derived these exact update rules from an information-theoretic outset.

[3] Derived from signal or time-series contexts, the instantaneousness of the model refers the assumption that the sources exclusively map to the sensors/mixtures at the same time instance (see chapter 3).

identical to ones applying to ICA, except for the crucial difference that, a priori, the sources are assumed to be *Gaussian*. From this formulation it is found that the sources can only be inferred up to a multiplication by a rotation matrix, that is, $\mathbf{s}_{\text{rot}} = \mathbf{U}\mathbf{s}$, where $\mathbf{U}$ is an orthogonal matrix. This is because the rotated source exhibit identical sufficient statistics

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \mathbf{A}\left\langle \mathbf{s}\mathbf{s}^\top \right\rangle \mathbf{A}^\top = \mathbf{A}\mathbf{A}^\top \tag{4.2}$$

$$\mathbf{A}\mathbf{U}^\top\mathbf{U}\left\langle \mathbf{s}\mathbf{s}^\top \right\rangle \mathbf{U}^\top\mathbf{U}\mathbf{A}^\top = \mathbf{A}\mathbf{A}^\top \tag{4.3}$$

where $\mathbf{s}$ is assumed to have zero mean and unit variance.

As a result, the PCA can estimate decorrelated components but not retrieve the sources of interest. In order to estimate the correct rotation of the sources, ICA methods exploit the hints provided by non-Gaussian distributions. In figure 4.1, the the rotation problem is illustrated for Gaussian sources versus uniformly distributed sources.

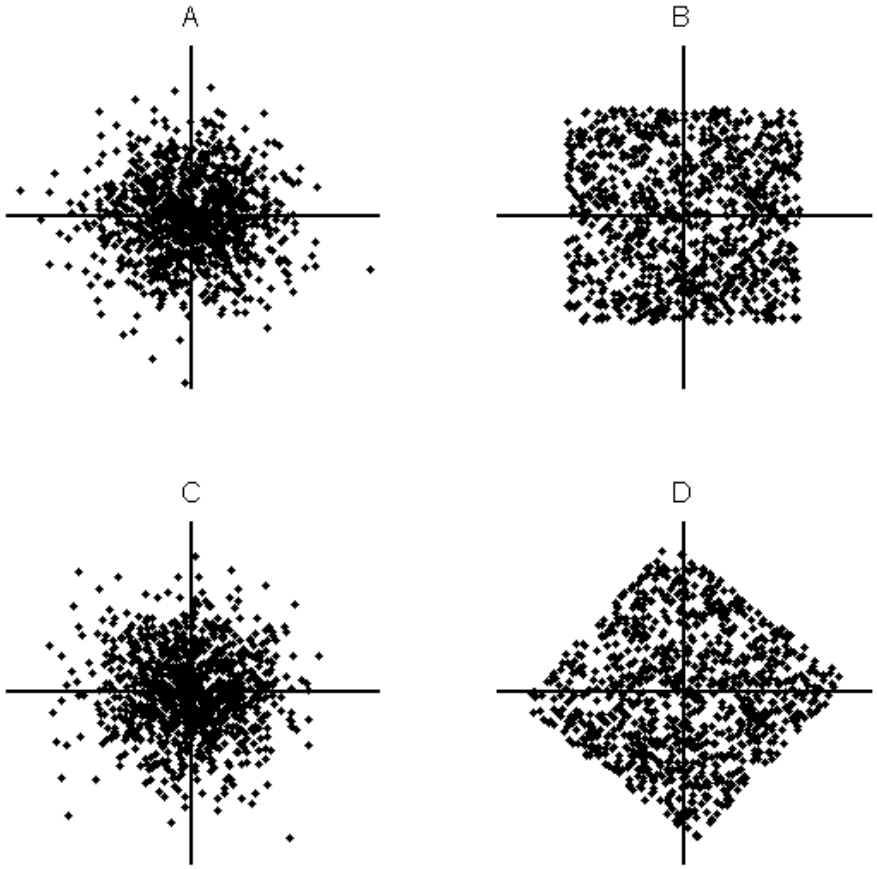

Figure 4.1: Cues provided by non-Gaussianity help identify sources in linear mixtures. The scatter plots show A) two Gaussian sources, B) two uniformly distributed sources. In C and D, the sources have been mixed by pre-multiplying with a rotation matrix. Whereas the Gaussian mixtures reveal no hints as to the correct de-rotation, this is not the case for the uniformly distributed sources. Reproduced from Hyvärinen et al. (2001).

Chapter 5

Conclusion

In this thesis is described a multitude of methods for source separation, employing a wide range of machine learning techniques as well as knowledge of speech and perception. In fact, a major feat of the author's contributions is the successful merger of fairly general models and specific audio domain models. In single-channel separation, the preprocessing was particularly important, since the sparse and non-negative factorizations are only viable in the time-frequency representation. The linear state-space model for multi-channel separation was augmented to contain a speech model, which may facilitate a faster adaptation to changes in the environment. Of course, the increased complexity of the models poses some additional challenges, namely the learning of the parameters and the inference of the sources. A great deal of research was devoted to overcoming these challenges, leading to an in-depth analysis of the expectation-maximization algorithm and stochastic/Newton-type gradient optimization.

An important lesson to draw is that, although the source separation problem can be formulated in very general terms, the solution cannot. The search for global solutions is tantamount to seeking an inverse for general systems. We should rather conciliate ourselves with the fact that there is not a single cure for 'mixedness', but rather a swarm of techniques that applies in different settings. The author's comments on two of the subproblems follow here.

5.1 Single-Channel Separation

The problem of separating more speakers from a microphone recording was treated in the first part of the thesis. On one hand it is difficult to perceive a meaningful mapping from a single dimension to many dimensions, but the operation is performed routinely by humans on a daily basis. This is the gold standard: to be able to perform on the level of humans, and it seems like we are getting closer. The research community has taken a big leap forward in the last few years with the application of advanced machine learning methods, such as the factorial hidden Markov model and new matrix factorization algorithms. Kristjansson et al. (2006) reported that their system outperformed humans in certain cases, measured in turns of word-error-rate on a recognition task.

The redundancy of speech plays a vital role, but also the detailed modelling of the speakers seems crucial to the results. Asari et al. (2006) make the argument that human perception also has library built-in sound models. However, it is an open problem to reduce the required amount of training data for learning the source-specific models. How to make the fullest use of psychoacoustic relations is another important question, specifically how to integrate information across time.

In this work, primarily speech was considered, but it is hugely interesting to extend the results to, *e.g.*, noise-removal. Schmidt et al. (2007) have taken the first steps in this direction, experimenting on wind noise.

5.2 Multi-Channel Separation

Despite a huge research effort, the problem of separating speech sources from convolutive mixtures is still largely unsolved. Consider as an example the level of flexibility which is available in mobile communication. It is possible to walk around with your cell-phone, in and out of buildings, even be a passenger in a car, and all the time, the transmission algorithms are tolerant to the changes in the signal path. At the same time the number of users (sources) in the network varies. These are features that would be necessary in order to use speech separation in, *e.g.*, conference room applications. But we do not have that, yet. The available algorithms require the signal channels to remain constant for seconds at a time, in

order to reach convergence of the filter coefficients. Also, the number of sources must be known in advance and remain constant, which is further unrealistic in applications.

Although I have found no magic bullet, I feel that some of the right ingredients have been presented in the thesis.

- The models are probabilistic or Bayesian up to a point, providing a framework for the incorporation of further priors on, *e.g.*, the filter taps. Additionally, it was demonstrated to be able to determine the correct model order on a sample (Olsson and Hansen, 2004a).

- A speech model was built into the framework, but more research is required to derive a practical algorithms. For example, a dynamic prior distribution could be formulated for the parameters of the speech model, *e.g.*, the fundamental frequency typically varies smoothly in time.

All in all, unanswered questions remain ...

Appendix I

B. A. Pearlmutter and R. K. Olsson, Algorithmic Differentiation of Linear Programs for Single-channel Source Separation, in proceedings of IEEE International Workshop on Machine Learning and Signal Processing, 2006

LINEAR PROGRAM DIFFERENTIATION FOR SINGLE-CHANNEL SPEECH SEPARATION

Barak A. Pearlmutter[*]

Hamilton Institute
National University of Ireland Maynooth
Co. Kildare, Ireland

Rasmus K. Olsson[†]

Informatics and Mathematical Modelling
Technical University of Denmark
2800 Lyngby, Denmark

ABSTRACT

Many apparently difficult problems can be solved by reduction to linear programming. Such problems are often subproblems within larger systems. When gradient optimisation of the entire larger system is desired, it is necessary to propagate gradients through the internally-invoked LP solver. For instance, when an intermediate quantity $\mathbf{z}$ is the solution to a linear program involving constraint matrix $\mathbf{A}$, a vector of sensitivities $dE/d\mathbf{z}$ will induce sensitivities $dE/d\mathbf{A}$. Here we show how these can be efficiently calculated, when they exist. This allows algorithmic differentiation to be applied to algorithms that invoke linear programming solvers as subroutines, as is common when using sparse representations in signal processing. Here we apply it to gradient optimisation of overcomplete dictionaries for maximally sparse representations of a speech corpus. The dictionaries are employed in a single-channel speech separation task, leading to 5 dB and 8 dB target-to-interference ratio improvements for same-gender and opposite-gender mixtures, respectively. Furthermore, the dictionaries are successfully applied to a speaker identification task.

1. INTRODUCTION

Linear programming solvers (LP) are often used as subroutines within larger systems, in both operations research and machine learning [1, 2]. One very simple example of this is in sparse signal processing, where it is common to represent a vector as sparsely as possible in an overcomplete basis; this representation can be found using LP, and the sparse representation is then used in further processing [3–9].

To date, it has not been practical to perform end-to-end gradient optimisation of algorithms of this sort. This is due to the difficulty of propagating intermediate gradients (adjoints) through the LP solver. We show below how these adjoint calculations can be done: how a sensitivity of the output can be manipulated to give a sensitivity of the inputs. As usual in Automatic Differentiation (AD), these do not require much more computation than the original primal LP calculation—in fact, rather unusually, here they may require considerably less.

We first introduce our notational conventions for LP, and then give a highly condensed introduction to, and notation for, AD. We proceed to derive AD transformations for a simpler subroutine than LP: a linear equation solver. (This novel derivation is of independent interest, as linear equations are often constructed and solved within larger algorithms.) Armed with a general AD transformation for linear equation solvers along with suitable notation, we find the AD transformations for linear program solvers simple to derive. This is applied mechanically to yield AD rules for a linearly-constrained L_1-optimiser.

The problem of finding an overcomplete signal dictionary tuned to a given stimulus ensemble, so that signals drawn from that ensemble will have sparse representations in the constructed dictionary, has received increasing attention, due to applications in both neuroscience and in the construction of efficient practical codes [10]. Here we derive a gradient method for such an optimisation, and apply it to learn a sparse representation of speech.

Single-channel speech separation, where the objective is to estimate the speech sources of the mixture, is a relevant task in hearing aids, as a speech recognition pre-processor, and in other applications which might benefit from better noise reduction. For this reason, there has been a flurry of interest in the problem [9, 11–17]. We encode the audio mixtures in the basis functions of the combined personalised dictionaries, which were adapted using the devised gradient method. The sparse code separates the signal into its sources, and reconstruction follows. Furthermore, we show that the dictionaries are truly personal, meaning that a given dictionary provides the sparsest fit for the particular speaker, which it was adapted to. Hence, we are able to correctly classify speech signals to their speaker.

[*]Supported by Science Foundation Ireland grant 00/PI.1/C067 and the Higher Education Authority of Ireland.

[†]Thanks to Oticon Fonden for financial support for this work.

2. BACKGROUND AND NOTATION

We develop a convenient notation while briefly reviewing the essentials of linear programming (LP) and algorithmic differentiation (AD).

2.1. Linear Programming

In order to develop a notation for LP, consider the general LP problem

$$\arg\min_{\mathbf{z}} \mathbf{w}^\top \mathbf{z} \text{ s.t. } \mathbf{A}\mathbf{z} \leq \mathbf{a} \text{ and } \mathbf{B}\mathbf{z} = \mathbf{b} \tag{1}$$

We will denote the linear program solver lp, and write the solution as $\mathbf{z} = \mathsf{lp}(\mathbf{w}, \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b})$. It is important to see that $\mathsf{lp}(\cdot)$ can be regarded as either a mathematical function which maps LP problems to their solutions, or as a computer program which actually solves LP problems. Our notation deliberately does not distinguish between these two closely related notions.

Assuming feasibility, boundedness, and uniqueness, the solution to this LP problem will satisfy a set of linear equalities consisting of a subset of the constraints: the *active* constraints [18–20]. An LP solver calculates two pieces of information: the solution itself, and the identity of the active constraints. We will find it convenient to refer to the active constraints by defining some very sparse matrices that extract the active constraints from the constraint matrices. Let $\alpha_1 < \cdots < \alpha_n$ be the indices of the rows of $\mathbf{A}$ corresponding to active constraints, and $\beta_1 < \cdots < \beta_m$ index the active rows of $\mathbf{B}$. Without loss of generality, we assume that the total number of active constraints is equal the dimensionality of the solution, $n + m = \dim \mathbf{z}$. We let $\mathbf{P}_\alpha$ be a matrix with n rows, where the i-th row is all zeros except for a one in the α_i-th column, and $\mathbf{P}_\beta$ similarly have m rows, with its i-th row all zeros except for a one in the β_i-th column. So $\mathbf{P}_\alpha\mathbf{A}$ and $\mathbf{P}_\beta\mathbf{B}$ hold the active rows of $\mathbf{A}$ and $\mathbf{B}$, respectively. These can be combined into a single matrix,

$$\mathbf{P} \equiv \begin{bmatrix} \mathbf{P}_\alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\beta \end{bmatrix}$$

Using these definitions, the solution $\mathbf{z}$ to (1), which presumably is already available having been computed by the algorithm that identified the active constraints, must be the unique solution of the system of linear constraints

$$\mathbf{P}\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\mathbf{z} = \mathbf{P}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}$$

or

$$\mathsf{lp}(\mathbf{w}, \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b}) = \mathsf{lq}(\mathbf{P}\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}, \mathbf{P}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}) \tag{2}$$

where lq is a routine that efficiently solves a system of linear equations, $\mathsf{lq}(\mathbf{M}, \mathbf{m}) = \mathbf{M}^{-1}\mathbf{m}$. For notational convenience we suppress the identity of the active constraints as an output of the lp routine. Instead we assume that it is available where necessary, so any function with access to the solution $\mathbf{z}$ found by the LP solver is also assumed to have access to the corresponding $\mathbf{P}$.

2.2. Algorithmic Differentiation

AD is a process by which a numeric calculation specified in a computer programming language can be mechanically transformed so as to calculate derivatives (in the differential calculus sense) of the function originally calculated [21]. There are two sorts of AD transformations: forward accumulation [22] and reverse accumulation [23]. (A special case of reverse accumulation AD is referred to as backpropagation in the machine learning literature [24].) If the entire calculation is denoted $\mathbf{y} = h(\mathbf{x})$, then forward accumulation AD arises because a perturbation $d\mathbf{x}/dr$ induces a perturbation $d\mathbf{y}/dr$, and reverse accumulation AD arises because a gradient $dE/d\mathbf{y}$ induces a gradient $dE/d\mathbf{x}$. The Jacobian matrix plays a dominant role in reasoning about this process. This is the matrix $\mathbf{J}$ whose i,j-th entry is dh_i/dx_j. Forward AD calculates $\acute{\mathbf{y}} = \mathbf{J}\acute{\mathbf{x}} = \overrightarrow{h}(\mathbf{x}, \acute{\mathbf{x}})$, and reverse AD calculates $\grave{\mathbf{x}} = \mathbf{J}^\top\grave{\mathbf{y}} = \overleftarrow{h}(\mathbf{x}, \grave{\mathbf{y}})$. The difficulty is that, in high dimensional systems, the matrix $\mathbf{J}$ is too large to actually calculate. In AD the above matrix-vector products are found directly and efficiently, without actually calculating the Jacobian.

The central insight is that calculations can be broken down into a chained series of assignments $v := g(u)$, and transformed versions of these chained together. The transformed version of the above internal assignment statement would be $\acute{v} := \overrightarrow{g}(u, \acute{u}, v)$ in forward mode [22], or $\grave{u} := \overleftarrow{g}(u, v, \grave{v})$ in reverse mode [23]. The most interesting property of AD, which results from this insight, is that the time consumed by the adjoint calculations can be the same as that consumed by the original calculation, up to a small constant factor. (This naturally assumes that the transformations of the primitives invoked also obey this property, which is in general true.)

We will refer to the adjoints of original variables introduced in forward accumulation (perturbations) using a forward-leaning accent $v \mapsto \acute{v}$; to the adjoint variables introduced in the reverse mode transformation (sensitivities) using a reverse-leaning accent $v \mapsto \grave{v}$; and to the forward- and reverse-mode transformations of functions using forward and reverse arrows, $h \mapsto \overrightarrow{h}$ and $h \mapsto \overleftarrow{h}$. A detailed introduction to AD is beyond the scope of this paper, but one form appears repeatedly in our derivations. This is $\mathbf{V} := \mathbf{A}\mathbf{U}\mathbf{B}$ where $\mathbf{A}$ and $\mathbf{B}$ are constant matrices and $\mathbf{U}$ and $\mathbf{V}$ are matrices as well. This transforms to $\acute{\mathbf{V}} := \mathbf{A}\acute{\mathbf{U}}\mathbf{B}$ and $\grave{\mathbf{U}} := \mathbf{A}^\top\grave{\mathbf{V}}\mathbf{B}^\top$.

2.3. AD of a Lin. Eq. Solver

We first derive AD equations for a simple implicit function, namely a linear equation solver. We consider a subroutine lq which finds the solution $\mathbf{z}$ of $\mathbf{M}\mathbf{z} = \mathbf{m}$, written $\mathbf{z} = \mathsf{lq}(\mathbf{M}, \mathbf{m})$. This assumes that $\mathbf{M}$ is square and full-rank, just as a division operation $z = x/y$ assumes that $y \neq 0$. We will derive formulae for both forward mode AD (the $\acute{\mathbf{z}}$ induced by $\acute{\mathbf{M}}$ and $\acute{\mathbf{m}}$) and reverse mode AD (the $\grave{\mathbf{M}}$ and $\grave{\mathbf{m}}$ induced by $\grave{\mathbf{z}}$).

For forward propagation of perturbations, we will write $\acute{\mathbf{z}} = \overrightarrow{\mathsf{lq}}(\mathbf{M}, \acute{\mathbf{M}}, \mathbf{m}, \acute{\mathbf{m}}, \mathbf{z})$. Because $(\mathbf{M} + \acute{\mathbf{M}})(\mathbf{z} + \acute{\mathbf{z}}) = \mathbf{m} + \acute{\mathbf{m}}$ which reduces to $\mathbf{M}\acute{\mathbf{z}} = \acute{\mathbf{m}} - \acute{\mathbf{M}}\mathbf{z}$, we conclude that

$$\overrightarrow{\mathsf{lq}}(\mathbf{M}, \acute{\mathbf{M}}, \mathbf{m}, \acute{\mathbf{m}}, \mathbf{z}) = \mathsf{lq}(\mathbf{M}, \acute{\mathbf{m}} - \acute{\mathbf{M}}\mathbf{z}).$$

Note that lq is linear in its second argument, where the perturbations enter linearly. For reverse propagation of sensitivities, we will write

$$\begin{bmatrix}\grave{\mathbf{M}} & \grave{\mathbf{m}}\end{bmatrix} = \overleftarrow{\mathsf{lq}}(\mathbf{M}, \mathbf{m}, \mathbf{z}, \grave{\mathbf{z}}). \tag{3}$$

First observe that $\mathbf{z} = \mathbf{M}^{-1}\mathbf{m}$ and hence $\grave{\mathbf{m}} = \mathbf{M}^{-\top}\grave{\mathbf{z}}$ so

$$\grave{\mathbf{m}} = \mathsf{lq}(\mathbf{M}^\top, \grave{\mathbf{z}}).$$

For the remaining term we start with our previous forward perturbation $\acute{\mathbf{M}} \mapsto \acute{\mathbf{z}}$, namely $\acute{\mathbf{z}} = -\mathbf{M}^{-1}\acute{\mathbf{M}}\mathbf{z}$, and note that the reverse must be the transpose of this linear relationship, $\grave{\mathbf{M}} = -\mathbf{M}^{-\top}\grave{\mathbf{z}}\mathbf{z}^\top$, which is the outer product

$$\grave{\mathbf{M}} = -\grave{\mathbf{m}}\mathbf{z}^\top.$$

2.4. AD of Linear Programming

We apply equation (3) followed by some bookkeeping, yields

$$\begin{aligned}\begin{bmatrix}\grave{\mathbf{A}} & \grave{\mathbf{a}}\\ \grave{\mathbf{B}} & \grave{\mathbf{b}}\end{bmatrix} &= \overleftarrow{\mathsf{lp}}(\mathbf{w}, \mathbf{A}, \mathbf{a}, \mathbf{B}, \mathbf{b}, \mathbf{z}, \grave{\mathbf{z}})\\ &= \mathbf{P}^\top \overleftarrow{\mathsf{lq}}(\mathbf{P}\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}, \mathbf{P}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \mathbf{z}, \grave{\mathbf{z}})\\ \grave{\mathbf{w}} &= \mathbf{0}\end{aligned}$$

Forward accumulation is similar, but is left out for brevity.

2.5. Constrained L_1 Optimisation

We can find AD equations for linearly constrained L_1-norm optimisation via reduction to LP. Consider

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ s.t. } \mathbf{D}\mathbf{c} = \mathbf{y}.$$

Although $\|\mathbf{c}\|_1 = \sum_i |c_i|$ is a nonlinear objective function, a change in parametrisation allows optimisation via LP. We name the solution $\mathbf{c} = \mathsf{L1opt}(\mathbf{y}, \mathbf{D})$ where

$$\mathsf{L1opt}(\mathbf{y}, \mathbf{D}) = \begin{bmatrix}\mathbf{I} & -\mathbf{I}\end{bmatrix} \mathsf{lp}(\mathbf{1}, -\mathbf{I}, \mathbf{0}, \mathbf{D}\begin{bmatrix}\mathbf{I} & -\mathbf{I}\end{bmatrix}, \mathbf{y})$$

in which $\mathbf{0}$ and $\mathbf{1}$ denote column vectors whose elements all contain the indicated number, and each $\mathbf{I}$ is an appropriately sized identity matrix. The reverse-mode AD transformation follows immediately,

$$\begin{aligned}&\overleftarrow{\mathsf{L1opt}}(\mathbf{y}, \mathbf{D}, \mathbf{c}, \grave{\mathbf{c}}) = \begin{bmatrix}\grave{\mathbf{D}} & \grave{\mathbf{y}}\end{bmatrix} =\\ &\begin{bmatrix}\mathbf{0}' & \mathbf{I}\end{bmatrix} \overleftarrow{\mathsf{lp}}(\mathbf{1}, -\mathbf{I}, \mathbf{0}, \mathbf{D}\begin{bmatrix}\mathbf{I} & -\mathbf{I}\end{bmatrix}, \mathbf{y}, \mathbf{z}, \begin{bmatrix}\mathbf{I}\\ -\mathbf{I}\end{bmatrix}\grave{\mathbf{c}}) \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathbf{I} & \mathbf{0}\\ \mathbf{0}^\top & 1\end{bmatrix}\end{aligned}$$

where $\mathbf{z}$ is the solution of the internal LP problem and $\mathbf{0}'$ is an appropriately sized matrix of zeros.

3. DICTIONARIES OPTIMISED FOR SPARSITY

A major advantage of the LP differentiation framework, and more specifically the reverse accumulation of the constrained L_1 norm optimisation, is that it provides directly a learning rule for learning sparse representation in overcomplete dictionaries.

We assume an overcomplete dictionary in the columns of $\mathbf{D}$, which is used to encode a signal represented in the column vector $\mathbf{y}$ using the column vector of coefficients $\mathbf{c} = \mathsf{L1opt}(\mathbf{y}, \mathbf{D})$ where each dictionary element has unit L_2 length. A probabilistic interpretation of the encoding as a maximum posterior (MAP) estimate naturally follows from two assumptions: a Laplacian prior $\mathrm{p}(\mathbf{c})$, and a noise-free observation model $\mathbf{y} = \mathbf{D}\mathbf{c}$. This gives

$$\mathbf{c} = \arg\max_{\mathbf{c}'} \mathrm{p}(\mathbf{c}'|\mathbf{y}, \mathbf{D})$$

We would like to improve $\mathbf{D}$ for a particular distribution of signals, meaning change $\mathbf{D}$ so as to maximise the sparseness of the codes assigned. With $\mathbf{y}$ drawn from this distribution, an ideal dictionary will minimise the average code length, giving maximally sparse coefficients. We will update $\mathbf{D}$ so as to minimise $E = \langle \|\mathsf{L1opt}(\mathbf{y}, \mathbf{D})\|_1 \rangle$ while keeping the columns of $\mathbf{D}$ at unit length. This can be regarded a special case of Independent Component Analysis [25], where measures of independence across coefficients are optimised. We wish to use a gradient method so we calculate $\nabla_{\mathbf{D}} E_{\mathbf{y}}$ where $E_{\mathbf{y}} = \|\mathsf{L1opt}(\mathbf{y}, \mathbf{D})\|_1$ making $E = \langle E_{\mathbf{y}} \rangle$. Invoking AD,

$$\begin{aligned}\nabla_{\mathbf{D}} E_{\mathbf{y}} = \grave{\mathbf{D}} &= \begin{bmatrix}\grave{\mathbf{D}} & \grave{\mathbf{y}}\end{bmatrix}\begin{bmatrix}\mathbf{I}\\ \mathbf{0}^\top\end{bmatrix}\\ &= \overleftarrow{\mathsf{L1opt}}(\mathbf{y}, \mathbf{D}, \mathbf{c}, \mathrm{sign}(\mathbf{c}))\begin{bmatrix}\mathbf{I}\\ \mathbf{0}^\top\end{bmatrix}\end{aligned} \tag{4}$$

where $\mathrm{sign}(x) = +1/0/-1$ for x positive/zero/negative, and applies elementwise to vectors.

We are now in a position to perform stochastic gradient optimisation [26], modified by the inclusion of a normalisation step to maintain the columns of $\mathbf{D}$ at unit length and non-negative.

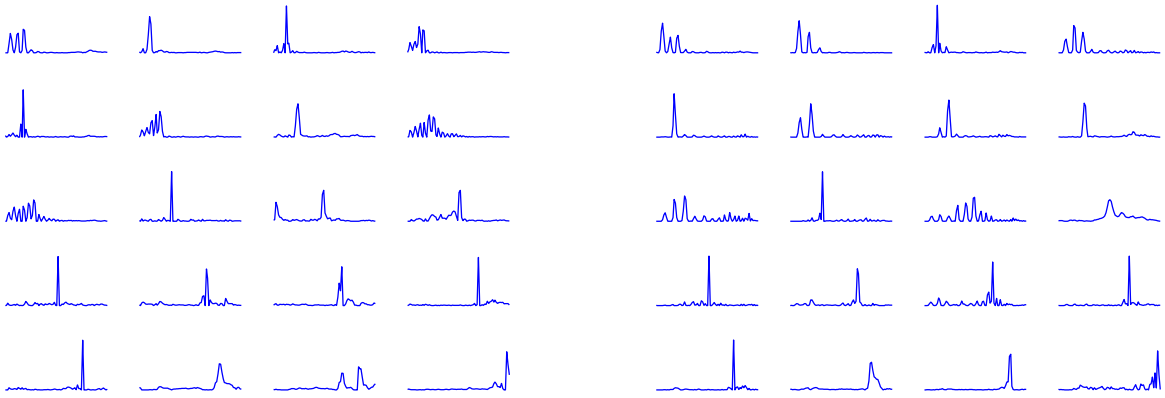

Fig. 1. A sample of learnt dictionary entries for male (left) and female (right) speech in the Mel spectrum domain. Clearly, harmonic features emerge from the data but some broad and narrow noise spectra can also be seen. The dictionaries were initialised to $N = 256$ delta-like pulses, length $L = 80$ and were adopted from $T = 420\,\mathrm{s}$ of speech.

1. Draw $\mathbf{y}$ from signal distribution.
2. Calculate $E_{\mathbf{y}}$.
3. Calculate $\nabla_{\mathbf{D}} E_{\mathbf{y}}$ by (4).
4. Step $\mathbf{D} := \mathbf{D} - \eta \nabla_{\mathbf{D}} E_{\mathbf{y}}$.
5. Set any negative element of $\mathbf{D}$ to zero.
6. Normalise the columns $\mathbf{d}_i$ of $\mathbf{D}$ to unit L_2 norm.
7. Repeat to convergence of $\mathbf{D}$.

This procedure can be regarded as a very efficient exact maximum likelihood treatment of the posterior integrated using a Gaussian approximation [7]. However, the formulation here can be easily and mechanically generalised to other objectives.

A set of personalised speech dictionaries were learnt by sparsity optimisation in the Grid Corpus [27] which is available at http://www.dcs.shef.ac.uk/spandh/gridcorpus. This corpus contains 1000×34 utterances of 34 speakers, confined to a limited vocabulary. The speech was preprocessed and represented to (essentially) transform the audio signals into a Mel time-frequency representation, as presented and discussed by Ellis and Weiss [14]. The data was downsampled to $8\,\mathrm{kHz}$ and high-pass filtered to bias our objective towards more accuracy in the high-end of the spectrum. The short-time Fourier transform was computed from windowed data vectors of length $32\,\mathrm{ms}$, corresponding to $K = 256$ samples, and subsequently mapped into $L = 80$ bands on the Mel scale. From $T = 420\,\mathrm{s}$ of audio from each speaker, the non-zero time-frames were extracted for training and normalised to unity L2 norm. The remainder of the audio ($> 420\,\mathrm{s}$) was set aside for testing. The stochastic gradient optimisation of the linearly constrained L_1 norm was run for 40,000 iterations. The step-size η was decreased throughout the training. The $N = 256$ columns of the dictionaries were initialised with narrow pulses distributed evenly across the spectrum and non-negativity was enforced following each iteration. In Figure 1 is displayed a randomly selected sample of learnt dictionary elements of one male and one female speaker. The dictionaries clearly capture a number of characteristics of speech, such as quasi-periodicity and dependencies across frequency bands.

3.1. Source Separation

This work was motivated by a particular application: single-channel source separation.[1] The aim is to recover R source signals from a one-dimensional mixture signal. In that context, an important technique is to perform a linearly constrained L_1-norm optimisation in order to fit an observed signal using a sparse subset of coefficients over an overcomplete signal dictionary. A single column of the mixture spectrogram is the sum of the source spectra: $\mathbf{y} = \sum_i^R \mathbf{y}_i$. In the interest of simplicity, this model assumes a $0\,\mathrm{dB}$ target-to-masker ratio (TMR). Generalization to general TMR by the inclusion of weighting coefficients is straightforward.

As a generative signal model, it is assumed that $\mathbf{y}_i$ can be represented sparsely in the overcomplete dictionary $\mathbf{D}$, which is the concatenation of the source dictionaries: $\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \dots & \mathbf{D}_i & \dots & \mathbf{D}_R \end{bmatrix}$. Assuming that the $\mathbf{D}_i$ are different in some sense, it can be expected that a sparse representation in the overcomplete basis $\mathbf{D}$ coincides with the separation of the sources, *i.e.* we compute

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1^\top & \dots & \mathbf{c}_i^\top & \dots & \mathbf{c}_R^\top \end{bmatrix}^\top = \mathsf{L1opt}(\mathbf{y}, \mathbf{D})$$

where the $\mathbf{c}_i$ are the coefficients pertaining to the ith source. The source estimates in the Mel spectrum domain are then re-synthesised as $\hat{\mathbf{y}}_i = \mathbf{D}_i \mathbf{c}_i$. The conversion back to the time-domain consists of mapping to the amplitude spectro-

[1] The INTERSPEECH 2006 conference hosts a special session on this issue, based on the GRID speech corpus. See www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.htm.

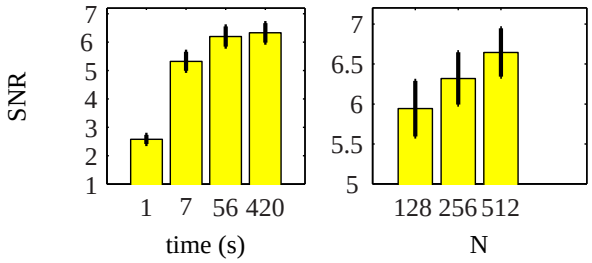

Fig. 2. Dependency of the separation performance measured as signal-to-noise ratio (SNR) as a function of the data volume (left), and, the dictionary size, N (right). Only $T = 7\,\mathrm{s}$ of speech is needed to attain near-optimal performance. The performance increases about 0.5 dB per doubling of N.

Genders	SNR (dB)
M/M	4.9 ± 1.2
M/F	7.8 ± 1.3
F/F	5.1 ± 1.4

Table 1. Monaural two-speaker signal-to-noise separation performance (mean$\pm$stderr of SNR), by speaker gender. The simulated test data consisted of all possible combinations, $T = 6\,\mathrm{s}$, of the 34 speakers.

gram and subsequently reconstructing the time-domain signal using the noisy phase of the mixture. Due to the sparsity of speech in the transformed domain, the degree of overlap of the sources is small, which causes the approximation to be fairly accurate. Useful software in this connection is available at http://www.ee.columbia.edu/~dpwe/. In the following, the quality of $\hat{\mathbf{y}}_i$ are evaluated in the time-domain simply as the ratio of powers of the target to reconstruction error, henceforth termed the signal-to-noise ratio (SNR).

In order to assess the convergence properties of the algorithm, the SNR was computed as a function of the amount of training data, see figure 2. It was found that useful results could be achieved with a few seconds of training data, whereas optimal performance was only obtained after a few minutes. It was furthermore investigated how the SNR varies as a function of the number of dictionary elements, N. Each doubling of N brings an improvement, indicating the potential usefulness of increased computing power. The above results were obtained by simulating all possible mixtures of 8 speakers (4 male, 4 female) at $0\,\mathrm{dB}$ and computing the SNR's on $6\,\mathrm{s}$ segments. Performance figures were computed on the complete data set of 34 speakers, amounting to 595 combinations, with $N = 256$ and $T = 420\,\mathrm{s}$; see Table 1. The test data is available at www2.imm.dtu.dk/~rko/single_channel.

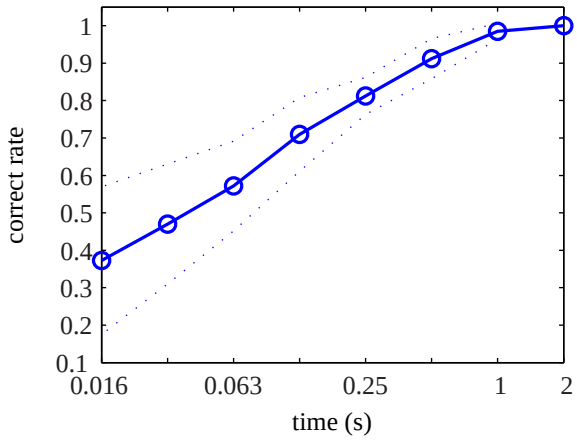

Fig. 3. The maximum-likelihood correct-classification rate as computed in a $T = 2\,\mathrm{s}$ test window on all combination of the 34 speakers and 34 dictionaries. If all time-frames are included into computation, the classification is perfect, but the performance decreases as smaller windows are used.

3.2. Speaker identification

In many potential applications of source separation the speakers of the mixture would be novel, and have to be estimated from the audio stream. In order to perform truly *blind* separation, the system should be able to automatically apply the appropriate dictionaries. Here we attack a simpler subproblem: speaker identification in an audio signal with only a single speaker. Our approach is straightforward: select the dictionary that yields the sparsest code for the signal. Again, this can be interpreted as maximum-likelihood classification. Figure 3 displays the percentage of correctly classified sound snippets. The figures were computed on all combinations of speakers and dictionaries, that is $34 \times 34 = 1156$ combinations. The complete data ($2\,\mathrm{s}$) resulted in all speakers being correctly identified. Shorter windows carried a higher error rate. For the described classification framework to be successful in a source separation task, it is required that each speaker appears exclusively in parts of the audio signal. This is not at all unrealistic in normal conversation, depending on the politeness of the speakers.

4. CONCLUSION AND OUTLOOK

Linear programming is often viewed as a black-box solver, which cannot be fruitfully combined with gradient-based optimisation methods. As we have seen, this is not the case. LP can be used as a subroutine in a larger system, and perturbations can be propagated forwards and sensitivities propagated backwards through the LP solver. The only caution is that LP is by nature only piecewise differentiable, so care must be taken with regard to crossing through such

discontinuities.

The figures carry evidence that the adapted Mel scale dictionaries to a large extent perform the job, and that the generalisation of the results to spontaneous speech depends to a large extent on designing a sensible scheme for providing the algorithm with a balanced training data. Furthermore, the system should be able to manage some difficult aspects of real-room conditions, in particular those in which the observed signal is altered by the room dynamics. We feel that a possible solution could build on the principles laid out in previous work [9], where a head-related transfer function (HRTF) is used to provide additional contrast between the sources.

We found that using spectrogram patches rather than power spectra improved the results only marginally, in agreement with previous reports using a related approach [16].

References

[1] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, July-Aug. 1995.

[2] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In *Adv. in Neu. Info. Proc. Sys. 9*, pages 368–374. MIT Press, 1997.

[3] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: A reweighted minimum norm algorithm. *IEEE Trans. Signal Proccessing*, 45(3):600–616, 1997.

[4] M. Lewicki and B. A. Olshausen. Inferring sparse, overcomplete image codes using an efficient coding framework. In *Advances in Neural Information Processing Systems 10*, pages 815–821. MIT Press, 1998.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[6] T.-W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters*, 4(5):87–90, 1999.

[7] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neu. Comp.*, 12(2):337–65, 2000.

[8] M. Zibulevsky and B. A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neu. Comp.*, 13(4):863–882, Apr. 2001.

[9] B. A. Pearlmutter and A. M. Zador. Monaural source separation using spectral cues. In *ICA*, pages 478–485, 2004.

[10] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neu. Comp.*, 15 (2):349–396, 2003.

[11] S. T. Roweis. One microphone source separation. In *Adv. in Neu. Info. Proc. Sys. 13*, pages 793–799. MIT Press, 2001.

[12] G.-J. Jang and T.-W. Lee. A maximum likelihood approach to single-channel source separation. *J. of Mach. Learn. Research*, 4:1365–1392, Dec. 2003.

[13] M. N. Schmidt and M. Mørup. Nonnegative matrix factor 2-D deconvolution for blind single channel source separation. In *ICA*, pages 123–123, 2006.

[14] D. P. W. Ellis and R. J. Weiss. Model-based monaural source separation using a vector-quantized phase-vocoder representation. In *ICASSP*, 2006.

[15] M. N. Schmidt and R. K. Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Interspeech*, 2006, submitted.

[16] S. T. Roweis. Factorial models and refiltering for speech separation and denoising. In *Eurospeech*, pages 1009–1012, 2003.

[17] F. Bach and M. I. Jordan. Blind one-microphone speech separation: A spectral learning approach. In *Advances in Neural Information Processing Systems 17*, pages 65–72, 2005.

[18] G. B. Dantzig. Programming in a linear structure. USAF, Washington D.C., 1948.

[19] S. I. Gass. *An Illustrated Guide to Linear Programming*. McGraw-Hill, 1970.

[20] R. Dorfman. The discovery of linear programming. *Annals of the History of Computing*, 6(3):283–295, July–Sep. 1984.

[21] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000. ISBN 0–89871–451–6.

[22] R. E. Wengert. A simple automatic derivative evaluation program. *Commun. ACM*, 7(8):463–464, 1964.

[23] B. Speelpenning. *Compiling Fast Partial Derivatives of Functions Given by Algorithms*. PhD thesis, Department of Computer Science, University of Illinois, Urbana-Champaign, Jan. 1980.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back–propagating errors. *Nature*, 323:533–536, 1986.

[25] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neu. Comp.*, 7(6):1129–1159, 1995.

[26] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Mat. Stats.*, 22:400–407, 1951.

[27] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition, 2005. Submitted.

Appendix II

M. N. Schmidt and R. K. Olsson, Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization, in proceedings of International Conference on Spoken Language Processing, 2006

Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization

Mikkel N. Schmidt and Rasmus K. Olsson

Informatics and Mathematical Modelling, Technical University of Denmark
mns,rko@imm.dtu.dk

Abstract

We apply machine learning techniques to the problem of separating multiple speech sources from a single microphone recording. The method of choice is a sparse non-negative matrix factorization algorithm, which in an unsupervised manner can learn sparse representations of the data. This is applied to the learning of personalized dictionaries from a speech corpus, which in turn are used to separate the audio stream into its components. We show that computational savings can be achieved by segmenting the training data on a phoneme level. To split the data, a conventional speech recognizer is used. The performance of the unsupervised and supervised adaptation schemes result in significant improvements in terms of the target-to-masker ratio.

Index Terms: Single-channel source separation, sparse non-negative matrix factorization.

1. Introduction

A general problem in many applications is that of extracting the underlying sources from a mixture. A classical example is the so-called cocktail-party problem in which the problem is to recognize or isolate what is being said by an individual speaker in a mixture of speech from various speakers. A particular difficult version of the cocktail-party problem occurs when only a single-channel recording is available, yet the human auditory system solves this problem for us. Despite its obvious possible applications in, e.g., hearing aids or as a preprocessor to a speech recognition system, no machine has been built, which solves this problem in general.

Within the signal processing and machine learning communities, the single channel separation problem has been studied extensively, and different parametric and non-parametric signal models have been proposed.

Hidden Markov models (HMM) are quite powerful for modelling a single speaker. It has been suggested by Roweis [1] to use a factorial HMM to separate mixed speech. Another suggestion by Roweis is to use a factorial-max vector quantizer [2]. Jang and Lee [3] use independent component analysis (ICA) to learn a dictionary for sparse encoding [4], which optimizes an independence measure across the encoding of the different sources. Pearlmutter and Olsson [5] generalize these results to overcomplete dictionaries, where the number of dictionary elements is allowed to exceed the dimensionality of the data. Other methods learn spectral dictionaries based on different types of non-negative matrix factorization (NMF) [6]. One idea is to assume a convolutive sum mixture, allowing the basis functions to capture time-frequency structures [7, 8].

A number researchers have taken ideas from the computational auditory scene analysis (CASA) literature, trying to incorporate various grouping cues of the human auditory system in speech separation algorithms [9, 10]. In the work by Ellis and Weiss [11] careful consideration is given to the representation of the audio signals so that the perceived quality of the separation is maximized.

In this work we propose to use the sparse non-negative matrix factorization (SNMF) [12] as a computationally attractive approach to sparse encoding separation. As a first step, overcomplete dictionaries are estimated for different speakers to give sparse representations of the signals. Separation of the source signals is achieved by merging the dictionaries pertaining to the sources in the mixture and then computing the sparse decomposition. We explore the significance of the degree of sparseness and the number of dictionary elements. We then compare the basic unsupervised SNMF with a supervised application of the same algorithm in which the training data is split into phoneme-level subproblems, leading to considerable computational savings.

2. Method

In the following, we consider modelling a magnitude spectrogram representation of a mixed speech signal. We represent the speech signal in the non-negative Mel spectrum magnitude domain, as suggested by Ellis and Weiss [11]. Here we posit that the spectrogram can be sparsely represented in an overcomplete basis,

$$\mathbf{Y} = \mathbf{D}\mathbf{H} \tag{1}$$

that is, each data point held in the columns of $\mathbf{Y}$ is a linear combination of few columns of $\mathbf{D}$. The dictionary, $\mathbf{D}$, can hold arbitrarily many columns, and the code matrix, $\mathbf{H}$, is sparse. Furthermore, we assume that the mixture signal is a sum of R source signals

$$\mathbf{Y} = \sum_{i}^{R} \mathbf{Y}_i.$$

The basis of the mixture signal is then the concatenation of the source dictionaries, $\mathbf{D} = [\mathbf{D}_1 \dots \mathbf{D}_i \dots \mathbf{D}_R]$, and the complete code matrix is the concatenation of the source-individual codes, $\mathbf{H} = \left[\mathbf{H}_1^\top \dots \mathbf{H}_i^\top \dots \mathbf{H}_R^\top\right]^\top$. By enforcing the sparsity of the code matrix, $\mathbf{H}$, it is possible to separate $\mathbf{Y}$ into its sources if the dictionaries are diverse enough.

As a consequence of the above, two connected tasks have to be solved: 1) the learning of source-specific dictionaries that yield sparse codes, and, 2) the computing of sparse decompositions for separation. We will use the sparse non-negative matrix factorization method proposed by Eggert and Körner [12] for both tasks.

2.1. Sparse Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) computes the decomposition in Equation (1) subject to the constraints that all matrices are non-negative, leading to solutions that are parts-based or sparse [6]. However, the basic NMF does not provide a well-defined solution in the case of overcomplete dictionaries, when the non-negativity constraints are not sufficient to obtain a sparse solution. The sparse non-negative matrix factorization (SNMF) optimizes the cost function

$$E = ||\mathbf{Y} - \bar{\mathbf{D}}\mathbf{H}||_F^2 + \lambda \sum_{ij} \mathbf{H}_{ij} \quad \text{s.t.} \quad \mathbf{D}, \mathbf{H} \geq \mathbf{0} \quad (2)$$

where $\bar{\mathbf{D}}$ is the column-wise normalized dictionary matrix. This cost function is the basic NMF quadratic cost augmented by an L_1 norm penalty term on the coefficients in the code matrix. The parameter, λ, controls the degree of sparsity. Any method that optimizes Equation (2) can be regarded as computing a maximum posterior (MAP) estimate given a Gaussian likelihood function and a one-sided exponential prior distribution over $\mathbf{H}$. The SNMF can be computed by alternating updates of $\mathbf{D}$ and $\mathbf{H}$ by the following rules [12]

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij} \bullet \frac{\mathbf{Y}_i^\top \bar{\mathbf{D}}_j}{\mathbf{R}_i^\top \bar{\mathbf{D}}_j + \lambda}$$

$$\mathbf{D}_j \leftarrow \mathbf{D}_j \bullet \frac{\sum_i \mathbf{H}_{ij} \left[\mathbf{Y}_i + (\mathbf{R}_i^\top \bar{\mathbf{D}}_j)\bar{\mathbf{D}}_j\right]}{\sum_i \mathbf{H}_{ij} \left[\mathbf{R}_i + (\mathbf{V}_i^\top \bar{\mathbf{D}}_j)\bar{\mathbf{D}}_j\right]}$$

where $\mathbf{R} = \mathbf{D}\mathbf{H}$, and the bold operators indicate pointwise multiplication and division.

We first apply SNMF to learn dictionaries of individual speakers. To separate speech mixtures we keep the dictionary fixed and update only the code matrix, $\mathbf{H}$. The speech is then separated by computing the reconstruction of the parts of the sparse decomposition pertaining to each of the used dictionaries.

2.2. Two Ways to Learn Sparse Dictionaries

We study two approaches to learning sparse dictionaries, see Figure 1. The first is a direct, unsupervised approach where the dictionary is learned by computing the SNMF on a large training data set of a single speaker. The second approach is to first segment the training data according to phoneme labels obtained by speech recognition software based on a hidden Markov model. Then, a sparse dictionary is learned for each phoneme and the final dictionary is constructed by concatenating the individual phoneme dictionaries. As a consequence, a smaller learning problem is addressed by the SNMF for each of the phonemes.

The computational savings associated with this divide-and-conquer approach are significant. Since the running time of the SNMF scales with the size of the training data and the number of elements in the dictionary, dividing the problem into SNMF subproblems for each phoneme reduces the overall computational burden by a factor corresponding to the number of phonemes. For example, if the data is split into 40 phonemes, we need to solve 40 SNMF subproblems each with a complexity of $1/40^2$ compared to the full SNMF problem. In addition to this, since the phoneme SNMF subproblems are much smaller than the total SNMF problem, a faster convergence of the iterative SNMF algorithm can be expected. These advantages makes it desirable to compare the quality of sparse dictionaries estimated by the two methods.

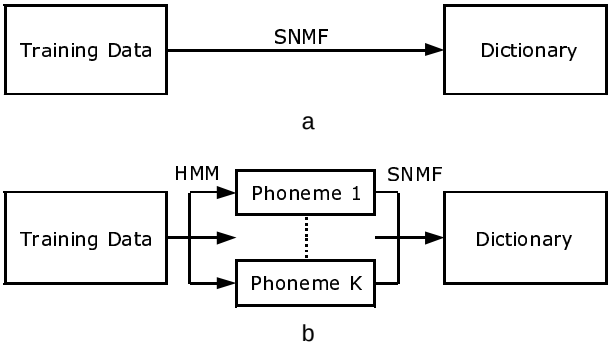

Figure 1: Two approaches for learning sparse dictionaries of speech. The first approach (a) is to learn the dictionary from a sparse non-negative matrix factorization of the complete training data. The second approach (b) is to segment the training data into individual phonemes, learn a sparse dictionary for each phoneme, and compute the dictionary by concatenating the individual phoneme dictionaries.

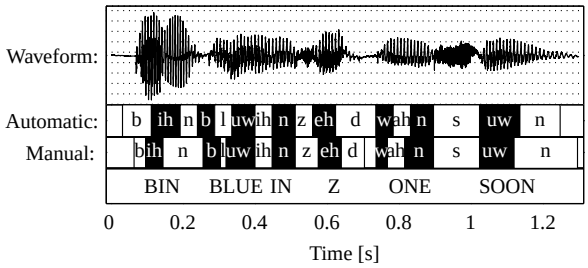

Figure 2: The automatic phoneme transcription as computed by the trained hidden Markov model (HMM) for an example sentence from the Grid Corpus. A manual transcription is provided for comparison, confirming the conventional hypothesis that the HMM is a useful tool in segmenting a speech signal into its phonemes.

3. Simulations

Part of the Grid Corpus [13] was used for evaluating the proposed method for speech separation. The Grid Corpus consists of simple structured sentences from a small vocabulary, and has 34 speakers and 1000 sentences per speaker. Each utterance is a few seconds and word level transcriptions are available. We used half of the corpus as a training set.

3.1. Phoneme Transcription

First, we used speech recognition software to generate phoneme transcriptions of the sentences. For each speaker in the corpus a phoneme-based hidden Markov model (HMM) was trained using the HTK toolkit[1]. The HMM's were used to compute an alignment of the phonemes in each sentence, taking the pronunciations of each word from the British English Example Pronunciation (BEEP) dictionary[2]. This procedure provided phoneme-level transcriptions of each sentence. In order to evaluate the quality

[1] Avaiable from htk.eng.cam.ac.uk.

[2] Available by anonymous ftp from svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/beep.tar.gz.

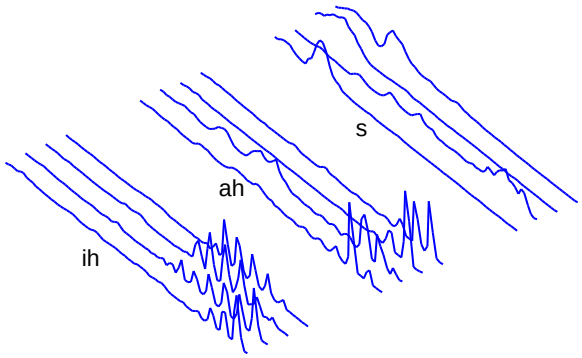

Figure 3: A few samples of columns of phoneme dictionaries learned from female speech. The SNMF was applied to data, which had been phoneme-labelled by a speech recognizer. Not surprisingly, the basis functions exhibit the some general properties of the respective phonemes, and additional variation is captured by the algorithm, such as the fundamental frequency in the case of voiced phonemes.

of the phoneme alignment, the automatic phoneme transcription was compared to a manual transcription for a few sentences. We found that the automatic phoneme alignment in general was quite reasonable. An example is given in Figure 2.

3.2. Preprocessing and Learning Dictionaries

We preprocessed the speech data in a similar fashion to Ellis and Weiss [11]: the speech was prefiltered with a high-pass filter, $1 - 0.95z^{-1}$, and the STFT was computed with an analysis window of 32ms at a sample rate of 25kHz. An overlap of 50 percent was used between frames. This yielded a spectrogram with 401 frequency bins which was then mapped into 80 frequency bins on the Mel scale. The training set was re-weighted so that all frames containing energy above a threshold were normalized by their standard deviation. The resulting magnitude Mel-scale spectrogram representation was employed in the experiments.

In order to assess the effects of the model hyper-parameters and the effect of splitting the training data according the phoneme transcriptions, a subset of four male and four female speakers were extracted from the Grid Corpus. We constructed a set of 64 mixed sentences by mixing two randomly selected sentences for all combinations of the eight selected test speakers.

Two different sets of dictionaries were estimated for each speaker. The first set was computed by concatenating the spectrograms for each speaker and computing the SNMF on the complete training data for that speaker. The second set was computed by concatenating the parts of the training data corresponding to each phoneme for each speaker, computing the SNMF for each phoneme spectrogram individually, and finally concatenating the individual phoneme dictionaries. To save computation, only 10 percent of the training set was used to train the dictionaries. In a Matlab environment running on a 1.6GHz Intel processor the computation of the SNMF for each speaker took approximately 30 minutes, whereas the SNMFs for individual phonemes were computed in a few seconds. The algorithm was allowed to run for maximally 500 iterations or until convergence as defined by the relative change in the cost function. Figure 3 shows

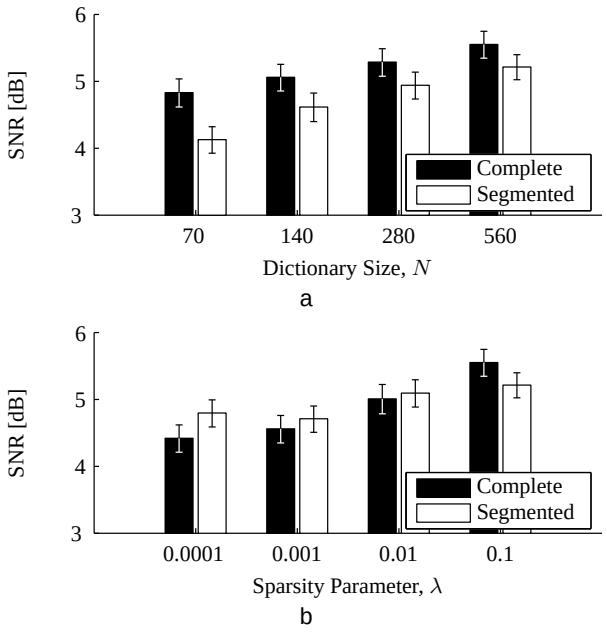

Figure 4: Average signal-to-noise ratio (SNR) of the separated signals for dictionaries trained on the complete speech spectrograms and on individual phonemes, (a) as a function of the dictionary size, N, with sparsity $\lambda = 0.1$, and (b) as a function of the sparsity with $N = 560$. We found that the SNMF algorithm did not give useful results when $\lambda = 1$.

samples from a dictionary which was learned using SNMF on the phoneme-segmented training data for a female speaker. The dictionaries were estimated for four different levels of sparsity, $\lambda = \{0.0001, 0.001, 0.01, 0.1\}$, and four different dictionary sizes, $N = \{70, 140, 280, 560\}$. This was done for both the complete and the phoneme-segmented training data.

3.3. Speech Separation

For each test sentence, we concatenated the dictionaries of the two speakers in the mixture, and computed the code matrix using the SNMF updates. Then, we reconstructed the individual magnitude spectra of the two speakers and mapped them from the Mel-frequency domain into the linear frequency STFT domain. Separated waveforms were computed by spectral masking and spectrogram inversion, using the original phase of the mixed signal. The separated waveforms were then compared with the original clean signals, computing the signal-to-noise ratio.

The results in Figure 4 show that the quality of separation increases with N. This agrees well with the findings of Ellis and Weiss [11]. Furthermore, the choice of sparsity, λ, is important for the performance of the separation method, especially in the case of unsegmented data. The individual phoneme-level dictionaries are so small in terms of N that the gain from enforcing sparsity in the NMF is not as significant; the segmentation in itself sparsifies the dictionary to some extend. Table 1 shows that the method works best for separating speakers of opposite gender, as would be expected. Audio examples are available at mikkelschmidt.dk/interspeech2006 .

	Complete	Segmented
Same gender	4.8 ± 0.4 dB	4.3 ± 0.3 dB
Opp. gender	6.6 ± 0.3 dB	6.4 ± 0.3 dB

Table 1: Average signal-to-noise ratio (SNR) of the separated signals for dictionaries trained on the complete speech spectrograms and on individual phonemes. Dictionaries were learned with $N = 560$ and $\lambda = 0.1$.

TMR	6dB	3dB	0dB	−3dB	−6dB	−9dB
Human Performance						
ST	90%	72%	54%	52%	60%	68%
SG	93%	85%	76%	72%	77%	80%
DG	94%	91%	86%	88%	87%	83%
All	92%	83%	72%	71%	75%	77%
Proposed Method						
ST	56%	53%	45%	38%	31%	28%
SG	60%	57%	52%	44%	37%	32%
DG	73%	72%	71%	63%	54%	41%
All	64%	62%	58%	51%	42%	35%

Table 2: Results from applying the SNMF to the Speech Separation Challenge: the word-recognition rate (WRR) on separated mixtures of speech in varying target-masker ratios (TMR) in same talker (ST), same gender (SG) different gender (DG), and overall (All) conditions compared with human performance on the mixtures. The WRR should be compared to that of other algorithms applied to the same test set (see the conference proceedings).

3.4. Interspeech 2006: Speech Separation Challenge

We evaluated the algorithm on the Speech Separation test set, which was constructed by adding a target and a masking speaker at different target-to-masker ratios (TMR)[3]. As an evaluation criterion, the word-recognition rate (WRR) for the letter and number in the target speech signal was computed using the HTK speech recognizer trained on data separated by the proposed method. A part of the test was to blindly identify the target signal as the one separated signal, which containing the word 'white'. A total of 600 mixtures were evaluated for each TMR. The source signals were separated and reconstructed in the time-domain as described previously. In Table 2, the performance of the method is contrasted with the performance of human listeners [14]. A subtask in obtaining these results was to estimate the identities of the speakers in the mixtures. This was done by exhaustively applying the SNMF to the signals with all pairs of two dictionaries, selecting the combination that gave the best fit. We are currently investigating methods to more efficiently determine the active sources in a mixture.

4. Discussion and Outlook

We have successfully applied sparse non-negative matrix factorization (SNMF) to the problem of monaural speech separation. The SNMF learns large overcomplete dictionaries, leading to a more sparse representations of individual speakers than for example the basic NMF. Inspection of the dictionaries reveals that they capture fundamental properties of speech, in fact they learn basis functions that resemble phonemes. This has lead us to adopt a working hypothesis that the learning of signal dictionaries on a phoneme level is a computational shortcut to the goal, leading to similar performance. Our experiments show that the practical performance of sparse dictionaries learned in this way performs only slightly worse than dictionaries learned on the complete dataset. In future work, we hope to benefit further from the phoneme labelling of the dictionaries in formulating transitional models in the encoding space of the SNMF, hopefully matching the dynamics of speech.

[3]This test set is due to Cooke and Lee. It is available at http://www.dcs.shef.ac.uk/ martin/SpeechSeparationChallenge.htm.

5. Acknowledgements

This work made possible in part by funding from Oticon Fonden. We would like to thank Lars Kai Hansen and Jan Larsen for fruitful discussions, and acknowledge Dan Ellis for making available useful software at his homepage.

6. References

[1] S. T. Roweis, "One microphone source separation," in *NIPS*, 2001, pp. 793–799.

[2] S. T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Eurospeech*, 2003, pp. 1009–1012.

[3] G. J. Jang and T. W. Lee, "A maximum likelihood approach to single channel source separation," *JMLR*, vol. 4, pp. 1365–1392, 2003.

[4] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[5] B. A. Pearlmutter and R. K. Olsson, "Algorithmic differentiation of linear programs for single-channel source separation," in *MLSP, submitted*, 2006.

[6] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[7] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *SAPA*, 2004.

[8] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *ICA*, 2005.

[9] B. A. Pearlmutter and A. M. Zador, "Monaural source separation using spectral cues," in *ICA*, 2004, pp. 478–485.

[10] F. Bach and M. I. Jordan, "Blind one-microphone speech separation: A spectral learning approach," in *NIPS*, 2005, pp. 65–72.

[11] D. P. W. Ellis and R. J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *ICASSP*, 2006.

[12] J. Eggert and E. Körner, "Sparse coding and nmf," in *Neural Networks*. 2004, vol. 4, pp. 2529–2533, IEEE.

[13] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *submitted to JASA*.

[14] M. P. Cooke, M. L. Garcia Lecumberri, and J. Barker, "The non-native cocktail party (in preparation)," .

Appendix III

M. N. Schmidt and R. K. Olsson, Feature Space Reconstruction for Single-Channel Speech Separation, in submission to Workshop on Applications of Signal Processing to Audio and Acoustics, 2007

Feature Space Reconstruction for Single-Channel Speech Separation

Mikkel N. Schmidt*, *Student Member, IEEE*,
Rasmus K. Olsson, *Student Member, IEEE*
Technical University of Denmark
Richard Petersens Plads, Bldg. 321
DK-2800 Kgs. Lyngby, Denmark
Email: mns@imm.dtu.dk
Fax: +45 45872599
Telephone: +45 45253888

Abstract

In this work we address the problem of separating multiple speakers from a single microphone recording. We formulate a linear regression model for estimating each speaker based on features derived from the mixture. The employed feature representation is a sparse, non-negative encoding of the speech mixture in terms of pre-learned speaker-dependent dictionaries. Previous work has shown that this feature representation by itself provides some degree of separation. We show that the performance is significantly improved when regression analysis is performed on the sparse, non-negative features.

Index Terms

Speech separation, single-channel, monaural, sparse non-negative matrix factorization.

I. INTRODUCTION

The cocktail-party problem can be defined as that of isolating or recognizing speech from an individual speaker in the presence of interfering speakers. An impressive feature of the human auditory system, this is essentially possible using only one ear, or, equivalently, listening to a mono recording of the mixture. It is an interesting and currently unsolved research problem to devise an algorithm which can mimic this ability.

A number of signal processing approaches have been based on learning speaker-dependent models on a training set of isolated recordings and subsequently applying a combination of these to the mixture. One possibility is to use a hidden Markov model (HMM) based on a Gaussian mixture model (GMM) for each speech source and combine these in a factorial HMM to separate a mixture [1]. Direct (naive) inference in such a model is not practical because of the dimensionality of the combined state space of the factorial HMM, necessitating some trick in order to speed up the computations. Roweis shows how to obtain tractable inference by exploiting the fact that in a log-magnitude time-frequency representation, the sum of speech signals is well approximated by the maximum. This is reasonable, since speech is sparsely distributed in the time-frequency domain. Recently, impressive results have been achieved by Kristjansson et al. [2] who devise an efficient method of inference that does not use the max-approximation. Based on a range of other approximations, they devise a complex system which in some situations exceeds human performance in terms of the error rate in a word recognition task.

Bach and Jordan [3] do not learn speaker dependent models but instead decompose a mixture by clustering the time-frequency elements according to a parameterized distance measure designed with the psychophysics of speech in mind. The algorithm is trained by learning the parameters of the distance measure from a training data set.

Another class of algorithms, here denoted ‘dictionary methods’, generally rely on learning a matrix factorization, in terms of a dictionary and its encoding for each speaker, from training data. The dictionary is a source dependent basis, and the method relies on the dictionaries of the sources in the mixture being sufficiently different. Separation of a mixture is obtained by computing the combined encoding using the concatenation of the source dictionaries. As opposed to the HMM/GMM based methods, this does not require a combinatorial search and leads to faster inference. Different matrix factorization methods can be conceived based on various a priori assumptions. For instance, independent component analysis and sparse decomposition, where the encoding is assumed to be sparsely distributed, have been proposed for single-channel speech separation [4], [5]. Another way to constrain the matrices is achieved through the

assumption of non-negativity [6], [7], which is especially relevant when modeling speech in a magnitude spectrogram representation. Sparsity and non-negativity priors have been combined in sparse, non-negative matrix factorization [8] and applied to music and speech separation tasks [9], [10], [11].

In this work, we formulate a linear regression model for separating a mixture of speech signals based on features derived from a real-valued time-frequency representation of the speech. As a set of features, we use the encodings pertaining to dictionaries learned for each speaker using sparse, non-negative matrix factorization. The resulting maximum posterior estimator is linear in the observed mixture features and has a closed-form solution. We evaluate the performance of the method on synthetic speech mixtures by computing the signal-to-error ratio, which is the simplest, arguably sufficient, quality measure [12].

II. Methodology

The problem is to estimate P speech sources from a single microphone recording,

$$y(t) = \sum_{i=1}^{P} y_i(t), \tag{1}$$

where $y(t)$ and $y_i(t)$ are the time-domain mixture and source signals respectively. The separation is computed in an approximately invertible time-frequency representation, $\boldsymbol{Y} = \mathrm{TF}\{y(t)\}$, where $\boldsymbol{Y}$ is a real-valued matrix with spectral vectors as columns.

A. Linear estimator

In the following we describe a linear model for estimating the time-frequency representations of the sources in a mixture based on features derived from the mixture. The linear model reads,

$$\boldsymbol{Y}_i = \boldsymbol{W}_i^\top (\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top) + \boldsymbol{m}_i \boldsymbol{1}^\top + \boldsymbol{N}, \tag{2}$$

where $\boldsymbol{Y}_i = \mathrm{TF}\{y_i(t)\}$ is the time-frequency representation of the i'th source, $\boldsymbol{W}_i$ is a matrix of weights, $\boldsymbol{X}$ is a feature matrix derived from $\boldsymbol{Y}$, $\boldsymbol{\mu}$ is the mean of the features, $\boldsymbol{m}_i$ is the mean of the i'th source and $\boldsymbol{N}$ is an additive noise term.

If we assume that the noise follows an i.i.d. normal distribution, $\mathrm{vec}(\boldsymbol{N}) \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I})$, and put an i.i.d. zero mean normal prior over the weights, $\mathrm{vec}(\boldsymbol{W}_i) \sim \mathcal{N}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I})$, the maximum posterior (MAP) estimator of the i'th source is given by

$$\hat{\boldsymbol{Y}}_i = \boldsymbol{\Gamma}_i \boldsymbol{\Sigma}^{-1} (\boldsymbol{X}^* - \boldsymbol{\mu}\boldsymbol{1}^\top) + \boldsymbol{m}_i \boldsymbol{1}^\top, \tag{3}$$

where $\boldsymbol{X}^*$ is the feature mapping of the test mixture $\boldsymbol{Y}^*$ and

$$\boldsymbol{\Gamma}_i = \big(\boldsymbol{Y}_i - \boldsymbol{m}_i\boldsymbol{1}^\top\big)\big(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top\big)^\top, \tag{4}$$

$$\boldsymbol{\Sigma} = \big(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top\big)\big(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top\big)^\top + \frac{\sigma_n^2}{\sigma_w^2}\boldsymbol{I}. \tag{5}$$

Here, $\boldsymbol{X}$ is a matrix with feature vectors computed on a training mixture with mean $\boldsymbol{\mu}$, and $\boldsymbol{Y}_i$ is the corresponding time-frequency representation of the source with mean $\boldsymbol{m}_i$. For a detailed derivation of the MAP estimator, see e.g. Rasmussen and Williams [13].

When an isolated recording is available for each of the speakers, it is necessary to construct the feature matrix, $\boldsymbol{X}$, from synthetic mixtures. One way to exploit the available data would be to generate mixtures, $\boldsymbol{X}$, such that all possible combinations of time-indices are represented. However, the number of sources and/or the number of available time-frames would be prohibitively large.

A feasible approximation can be found in the limit of a large training set by making two additional assumptions: i) the features are additive, $\boldsymbol{X} = \sum_i^P \boldsymbol{X}_i$ with means $\boldsymbol{\mu}_i$, which is reasonable for, e.g., sparse features, and ii) the sources are independent such that all cross-products are negligible. Then,

$$\boldsymbol{\Gamma}_i \approx \big(\boldsymbol{Y}_i - \boldsymbol{m}_i\boldsymbol{1}^\top\big)\big(\boldsymbol{X_i} - \boldsymbol{\mu}_i\boldsymbol{1}^\top\big)^\top, \tag{6}$$

$$\boldsymbol{\Sigma} \approx \sum_{i=1}^{P}\big(\boldsymbol{X}_i - \boldsymbol{\mu}_i\boldsymbol{1}^\top\big)\big(\boldsymbol{X}_i - \boldsymbol{\mu}_i\boldsymbol{1}^\top\big)^\top. \tag{7}$$

B. Features

In this work, two sets of feature mappings are explored. The first, and most simple, is to use the time-frequency representation itself as input to the linear model,

$$\boldsymbol{X}_i = \boldsymbol{Y}_i, \qquad \boldsymbol{X}^* = \boldsymbol{Y}^*. \tag{8}$$

A second, more involved, possibility is to use the encodings of a sparse, non-negative matrix factorization algorithm (SNMF) [8] as the features (see appendix A for a summary of SNMF). Possibly, other dictionary methods provide equally viable features.

In the SNMF method, the time-frequency representation of the i'th source is modelled as $\boldsymbol{Y}_i \approx \boldsymbol{D}_i\boldsymbol{H}_i$ where $\boldsymbol{D}_i$ is a dictionary matrix containing a set of spectral basis vectors, and $\boldsymbol{H}_i$ is an encoding which describes the amplitude of each basis vector at each time point. In order to use the method to compute features for a mixture, a dictionary matrix is first learned separately on a training set for each of the sources. Next, the mixture and the training data is mapped onto the concatenated dictionaries of the sources,

$$\boldsymbol{Y}_i \approx \boldsymbol{D}\boldsymbol{H}_i, \qquad \boldsymbol{Y}^* \approx \boldsymbol{D}\boldsymbol{H}^*, \tag{9}$$

where $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_P]$. The encoding matrices, $\boldsymbol{H}_i$ and $\boldsymbol{H}^*$, are used as features,

$$\boldsymbol{X}_i = \boldsymbol{H}_i, \qquad \boldsymbol{X}^* = \boldsymbol{H}^*. \tag{10}$$

In previous work, the sources were estimated directly from these features as $\hat{\boldsymbol{Y}}_i = \boldsymbol{D}_i \boldsymbol{H}_i^*$ [11]. For comparison, we include this method in our evaluations. This method yields very good results when the sources, and thus the dictionaries, are sufficiently different from each other. In practice, however, this will not always be the case. In the factorization of the mixture, $\boldsymbol{D}_1$ will not only encode $\boldsymbol{Y}_1$ but also $\boldsymbol{Y}_2$ etc. This indicates that the encodings should rather be used as features in an estimator for each source.

III. Evaluation

The proposed speech separation method was evaluated on a subset of the GRID speech corpus [14] consisting of the first 4 male and first 4 female speakers (no. 1, 2, 3, 4, 5, 7, 11, and 15). The data was preprocessed by concatenating $T = 300$ s of speech from each speaker and resampling to $F_s = 8$ kHz. As a measure of performance, the signal-to-error ratio (SER) averaged across sources was computed in the time-domain. The testing was performed on synthetic 0 dB mixtures of two speakers, $T_{\text{test}} = 20$ s, constructed from all combinations of speakers in the test set.

In figures 1 and 2, the performance is shown for a collection of feature sets. The acronyms MAP-mel and MAP-SNMF refer to using the mel spectrum or the SNMF encoding as features, respectively. For reference, figures are provided for the basic SNMF approach as well [11]. The numeral suffix, '1' or '5', indicates whether using one or stacking five consecutive feature vectors, spaced 32 ms. The best performance is achieved for MAP-SNMF-5, reaching an $\simeq 1.2$ dB average improvement over the SNMF algorithm. It is noteworthy that the improvement is larger for the most difficult mixtures, those involving same-gender speakers.

In order to verify that the method is robust to changes in the relative gain of the signals in the mixtures, the performance was evaluated in a range of different target-to-interference ratios (TIR) (see figure 3). The results indicate that the method works very well even when the TIR is not known a priori. In figure 5, the performance is measured as a function of the available training data, indicating that the method is almost converged at 300 s.

The time-frequency representation was computed by normalizing the time-signals to unit power and computing the short-time Fourier transform (STFT) using 64 ms Hamming windows with 50% overlap. The absolute value of the STFT was then mapped onto a mel frequency scale using a publicly available toolbox [15] in order to reduce the dimensionality. Finally, the mel-frequency spectrogram was amplitude-compressed by exponentiating to the power p. By cross-validation we found that best results were obtained

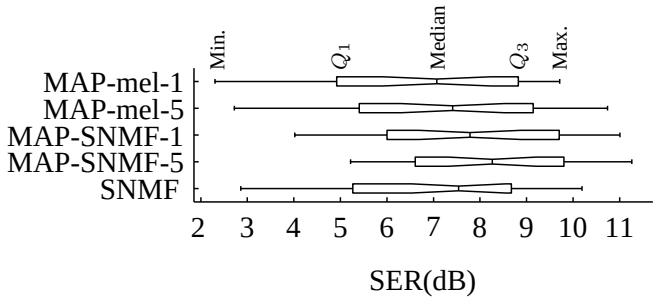

Fig. 1. The distribution of the signal-to-error (SER) performance of the method for all combinations of two speakers. The mel magnitude spectrogram (MAP-mel) and the SNMF encodings (MAP-SNMF) were used as features to the linear model. The results of using basic SNMF are given as a reference. The box plots indicate the extreme values along with the quartiles of the dB SER, averaged across sources.

at $p = 0.55$ which gave significantly better results compared with, e.g., operating in the amplitude ($p = 1$) or the power ($p = 2$) domains (see figure 4). Curiously, this model prediction is similar to the empirically determined $p \approx 0.67$ exponent used in power law modelling of perceived loudness in humans, known as Stevens' Law, (see for example Hermansky [16]).

In the dictionary learning phase, the SNMF algorithm was allowed 250 iterations to converge from random initial conditions drawn from a uniform distribution on the unit interval. The number of dictionary atoms was fixed at $r = 200$ and the level of sparsity was chosen by cross-validation to $\lambda = 0.15$. When computing the encodings on the test mixtures, we found that non-negativity alone was sufficiently restrictive, hence $\lambda = 0$.

Time-domain reconstruction was performed by binary masking in the STFT spectrogram and subsequent inversion using the phase of the original mixture as described for example by Wang and Brown [17]. The phase errors incurred by this procedure are not severe due to the sparsity of speech in the spectrogram representation. Audio examples of the reconstructed speech are available online [18].

IV. Discussion

The presented framework enjoys at least two significant advantages. First and foremost, computation in the linear model is fast. The estimation of the separation matrix is closed-form given the features, and the most time-consuming operation in the separation phase is a matrix product scaling with the dimensions of spectrogram and the number of features. Secondly, it is possible to fuse different features sets. Here,

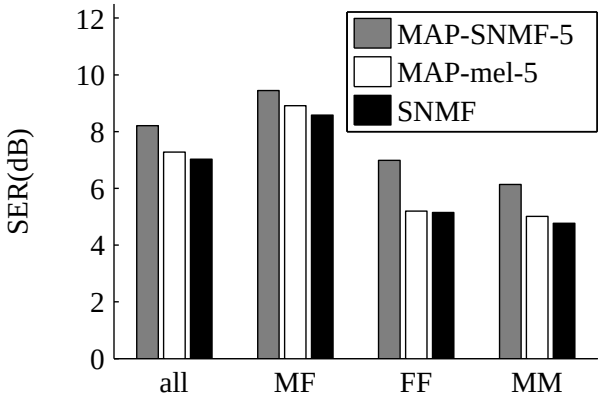

Fig. 2. The performance of the methods given as signal-to-error (SER) in dB, depending on the gender of the speakers. Male and female are identified by 'M' and 'F', respectively. The improvement of MAP-SNMF-5 over MAP-mel-5 and SNMF is largest in the most difficult (same-gender) mixtures.

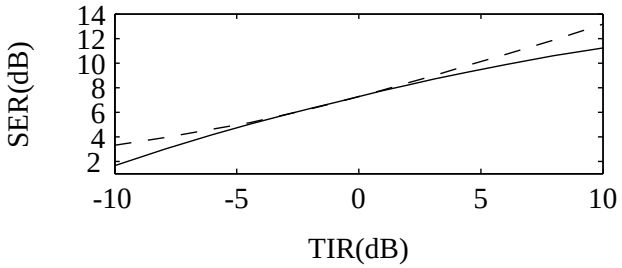

Fig. 3. The performance of the MAP-mel-5 algorithm given as the signal-to-error ratio (SER) of the target signal versus the target-to-interference ratio (TIR) of the mixture. The solid and dashed curves represent training on 0dB or the actual TIR of the test mixture, respectively. Clearly, the method is robust to a mismatch of the TIR between the training and test sets.

the spectrogram and sparse NMF were used, but many others could be imagined, possibly inspired by auditory models. The estimator integrates features across time, although the effect is relatively small, confirming previous reports that the inclusion of a dynamical model yields only marginal improvements [2], [19].

ACKNOWLEDGMENT

During the research process, L. K. Hansen, J. Larsen and O. Winther administered advice and good suggestions.

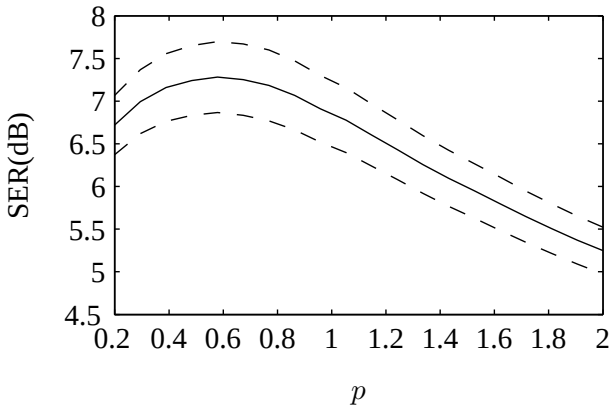

Fig. 4. The effect of amplitude compression on the performance of the MAP-mel-5 algorithm as measured in the signal-to-error ratio (SER). The optimal value of the exponent was found at $p \simeq 0.55$, in approximate accordance with Steven's power law for hearing. The dashed curve indicates the standard deviation of the mean.

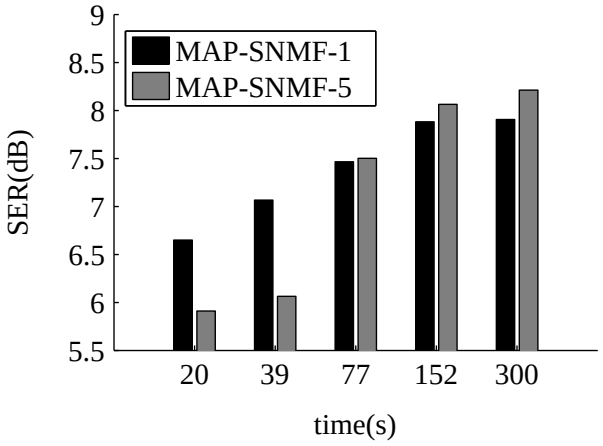

Fig. 5. The learning curve of the method, measured in signal-to-error ratio (SER), as a function of the size of the training set, depending on the complexity of the method.

APPENDIX

A. Sparse Non-negative Matrix Factorization

Let $\boldsymbol{Y} \geq \boldsymbol{0}$ be a non-negative data matrix. We model $\boldsymbol{Y}$ by

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{H} + \boldsymbol{N}, \tag{11}$$

where $\boldsymbol{N}$ is normal i.i.d. zero mean with variance σ_n^2. This gives rise to the likelihood function,

$$p(\boldsymbol{Y}|\boldsymbol{D},\boldsymbol{H}) \propto \exp\left(-\frac{|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{H}|_F^2}{2\sigma_n^2}\right), \tag{12}$$

where $|\cdot|_F$ denotes the Frobenius norm. We put a prior on $\boldsymbol{D}$ that is uniform over the part of the unit hyper-sphere lying in the positive orthant, i.e., $\boldsymbol{D}$ is non-negative and column-wise normalized. To obtain sparsity, the prior on $\boldsymbol{H}$ is assumed i.i.d. one-sided exponential, $p(\boldsymbol{H}) \propto \exp(-\beta|\boldsymbol{H}|_1)$, $\boldsymbol{H} \geq \boldsymbol{0}$, where $|\boldsymbol{H}|_1 = \sum_{ji}|h_{ji}|$. Now, the log-posterior can be written as

$$\log p(\boldsymbol{D},\boldsymbol{H}|\boldsymbol{Y}) \propto -\frac{1}{2}|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{H}|_F^2 - \lambda|\boldsymbol{H}|_1, \tag{13}$$
$$\text{s.t. } \boldsymbol{D} \geq \boldsymbol{0},\ |\boldsymbol{d}_j|_2 = 1,\ \boldsymbol{H} \geq \boldsymbol{0},$$

where $\boldsymbol{d}_j$ is the j'th column vector of $\boldsymbol{D}$.

The log-posterior can be seen as a quadratic cost function augmented by an L_1 norm penalty term on the coefficients in $\boldsymbol{H}$. The hyper-parameter $\lambda = \beta\sigma_n^2$ controls the degree of sparsity. A maximum posterior (MAP) estimate can be computed by optimizing (13) with respect to $\boldsymbol{D}$ and $\boldsymbol{H}$.

Eggert and Körner [8] derive a simple algorithm for computing this MAP estimate based on alternating multiplicative updates of $\boldsymbol{D}$ and $\boldsymbol{H}$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \bullet \frac{\bar{\boldsymbol{D}}^\top \boldsymbol{Y}}{\bar{\boldsymbol{D}}^\top \widetilde{\boldsymbol{Y}} + \boldsymbol{\Lambda}}, \tag{14}$$

$$\boldsymbol{d}_j \leftarrow \bar{\boldsymbol{d}}_j \bullet \frac{\sum_i h_{ji}\left[\boldsymbol{y}_i + (\widetilde{\boldsymbol{y}}_i^\top \bar{\boldsymbol{d}}_j)\bar{\boldsymbol{d}}_j\right]}{\sum_i h_{ji}\left[\widetilde{\boldsymbol{y}}_i + (\boldsymbol{y}_i^\top \bar{\boldsymbol{d}}_j)\bar{\boldsymbol{d}}_j\right]}, \tag{15}$$

where $\widetilde{\boldsymbol{Y}} = \bar{\boldsymbol{D}}\boldsymbol{H}$, $\bar{\boldsymbol{D}}$ is the column-wise normalized dictionary matrix, $\boldsymbol{\Lambda}$ is a matrix with elements λ, and the bold operators indicate pointwise multiplication and division.

REFERENCES

[1] S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems*, 2000, pp. 793–799.

[2] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, "Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006, pp. 97–100.

[3] F. R. Bach and M. I. Jordan, "Blind one-microphone speech separartion: A spectral learning approach," in *Advances in Neural Information Processing Systems*, 2005, pp. 65–72.

[4] G. J. Jang and T. W. Lee, "A maximum likelihood approach to single channel source separation," *Journal of Machine Learning Research*, vol. 4, pp. 1365–1392, 2003.

[5] B. A. Pearlmutter and R. K. Olsson, "Algorithmic differentiation of linear programs for single-channel source separation," in *Machine Learning and Signal Processing, IEEE International Workshop on*, 2006.

[6] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[7] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *Statistical and Perceptual Audio Processing (SAPA)*, 2004.

[8] J. Eggert and E. Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, vol. 4, 2004, pp. 2529–2533.

[9] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *International Computer Music Conference, ICMC*, 2003.

[10] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, "Non negative sparse representation for wiener based source separation with a single sensor," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 2003, pp. 613–616.

[11] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006.

[12] D. Ellis, "Evaluating speech separation systems," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Kluwer Academic Publishers, ch. 20, pp. 295–304.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[14] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *submitted to JASA*.

[15] D. P. W. Ellis. (2005) PLP and RASTA (and MFCC, and inversion) in Matlab. [Online]. Available: http://www.ee.columbia.edu/∼dpwe/resources/matlab/rastamat

[16] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[17] D. L. Wang and G. J. Brown, "Separation of speech from interfering sounds based on oscillatory correlation," *IEEE-NN*, vol. 10, no. 3, p. 684, 1999.

[18] M. N. Schmidt and R. K. Olsson. (2007) Audio samples relevant to this letter. [Online]. Available: http://mikkelschmidt.dk/spletters2007

[19] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transaction on Audio, Speech and Language Processing - to appear*, 2007.

Appendix IV

R. K. Olsson and L. K. Hansen, Probabilistic Blind Deconvolution of Non-stationary Sources, in proceedings of European Signal Processing Conference, 1697-1700, 2004

PROBABILISTIC BLIND DECONVOLUTION OF NON-STATIONARY SOURCES

Rasmus Kongsgaard Olsson and Lars Kai Hansen

Informatics and Mathematical Modelling, B321 Technical University of Denmark
DK-2800 Lyngby, Denmark
email: rko@isp.imm.dtu.dk,lkh@imm.dtu.dk

ABSTRACT

We solve a class of blind signal separation problems using a constrained linear Gaussian model. The observed signal is modelled by a convolutive mixture of colored noise signals with additive white noise. We derive a time-domain EM algorithm 'KaBSS' which estimates the source signals, the associated second-order statistics, the mixing filters and the observation noise covariance matrix. KaBSS invokes the Kalman smoother in the E-step to infer the posterior probability of the sources, and one-step lower bound optimization of the mixing filters and noise covariance in the M-step. In line with (Parra and Spence, 2000) the source signals are assumed time variant in order to constrain the solution sufficiently. Experimental results are shown for mixtures of speech signals.

1. INTRODUCTION

Reconstruction of temporally correlated source signals observed through noisy, convolutive mixtures is a fundamental theoretical issue in signal processing and is highly relevant for a number of important signal processing applications including hearing aids, speech processing, and medical imaging. A successful current approach is based on simultaneous diagonalization of multiple estimates of the source cross-correlation matrix [5]. A basic assumption in this work is that the source cross-correlation matrix is time variant. The purpose of the present work is to examine this approach within a probabilistic framework, which in addition to estimation of the mixing system and the source signals will allow us to estimate noise levels and model likelihoods.

We consider a noisy convolutive mixing problem where the sensor input $\mathbf{x}_t$ at time t is given by

$$\mathbf{x}_t = \sum_{k=0}^{L-1} \mathbf{A}_k \mathbf{s}_{t-k} + \mathbf{n}_t. \quad (1)$$

The L matrices $\mathbf{A}_k$ define the delayed mixture and $\mathbf{s}_t$ is a vector of possibly temporally correlated source processes. The noise $\mathbf{n}_t$ is assumed i.i.d. normal. The objective of blind source separation is to estimate the sources, the mixing parameters, and the parameters of the noise distribution.

Most blind deconvolution methods are based on higher-order statistics, see e.g. [4], [1]. However, the approach is proposed by Parra and Spence [5] is based on second order statistics and is attractive for its relative simplicity and implementation, yet excellent performance. The Parra and Spence algorithm is based on estimation of the inverse mixing process which maps measurements to source signals. A heuristic second order correlation function is minimized by the adaptation of the inverse process. The scheme needs multiple correlation measurements to obtain a unique inverse. This can be achieved, e.g., if the source signals are non-stationary or if the correlation functions are measured at time lags less than the correlation length of the source signals.

The main contribution of the present work is to provide an explicit statistical model for the decorrelation of convolutive mixtures of non-stationary signals. As a result, all parameters including mixing filter coefficients, source signal parameters and observation noise covariance are estimated by maximum-likelihood and the *exact* posterior distribution of the sources is obtained. The formulation is rooted in the theory of linear Gaussian models, see e.g., the review by Ghahramani and Roweis in [7]. The so-called Kalman Filter model is a state space model that can be set up to represent convolutive mixings of statistically independent sources added with observation noise. The standard estimation scheme for the Kalman filter model is an EM-algorithm that implements maximum-likelihood (ML) estimation of the parameters and maximum-posterior (MAP) inference of the source signals, see e.g. [3]. The specialization of the Kalman Filter model to convolutive mixtures is covered in section 2 while the adaptation of the model parameters is described in section 3. An experimental evaluation on a speech mixture is presented in section 4.

2. THE MODEL

The Kalman filter model is a generative dynamical state-space model that is typically used to estimate unobserved or hidden variables in dynamical systems, e.g. the velocity of an object whose position we are tracking. The basic Kalman filter model (no control inputs) is defined as

$$\begin{aligned} \mathbf{s}_t &= \mathbf{F}\mathbf{s}_{t-1} + \mathbf{v}_t \\ \mathbf{x}_t &= \mathbf{A}\mathbf{s}_t + \mathbf{n}_t \end{aligned} \quad (2)$$

The observed d_x-dimensional mixture, $\mathbf{x}_t = [x_{1,t}, x_{2,t}, .., x_{d_x,t}]^T$, is obtained from the multiplication of the mixing matrix, $\mathbf{A}$, on $\mathbf{s}_t$, the hidden state. The source innovation noise, $\mathbf{v}_t$, and the evolution matrix, $\mathbf{F}$, drive the sources. The signals are distributed as $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{s}_1 \sim \mathcal{N}(\mu, \mathbf{\Sigma})$.

By requiring $\mathbf{F}, \mathbf{Q}$ and $\mathbf{\Sigma}$ to be diagonal matrices, equation (2) satisfies the fundamental requirement of

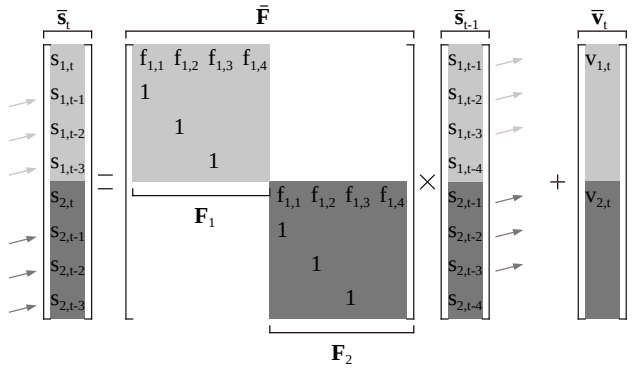

Figure 1: The AR(4) source signal model. The memory of $\mathbf{s}_t$ is updated by discarding $s_{i,t-4}$ and composing new $\mathbf{s}_{1,t}$ and $\mathbf{s}_{2,t}$ using the AR recursion. Blanks signify zeros.

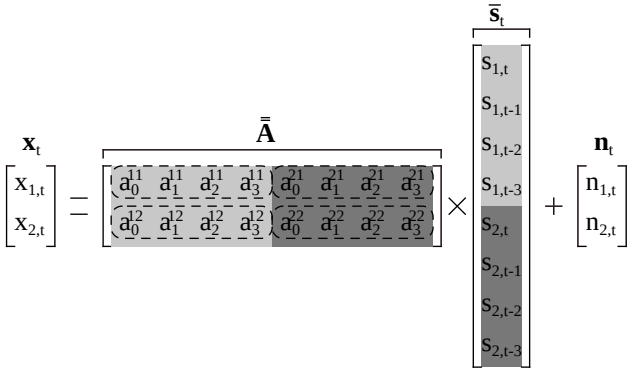

Figure 2: The convolutive mixing model requires a full $\bar{\bar{\mathbf{A}}}$ to be estimated.

any ICA formulation, namely that the sources are statistically independent. Under the diagonal constraint, this source model is identical to an AR(1) random process. In order for the Kalman model to be useful in the context of convolutive ICA for general temporally correlated sources we need to generalize it in two aspects, firstly we will move to higher order AR processes by stacking the state space, secondly we will introduce convolution in the observation model.

2.1 Model generalization

By generalizing (2) to AR(p) source models we can model wider classes of signals, including speech. The AR(p) model for source i is defined as:

$$s_{i,t} = f_{i,1}s_{i,t-1} + f_{i,2}s_{i,t-2} + .. + f_{i,p}s_{i,t-p} + v_{i,t}. \quad (3)$$

In line with e.g. [2], we implement the AR(p) process in the basic Kalman model by stacking the variables and parameters to form the augmented state vector

$$\bar{\mathbf{s}}_t = \left[\begin{array}{cccc} \mathbf{s}_{1,t}^T & \mathbf{s}_{2,t}^T & .. & \mathbf{s}_{d_s,t}^T \end{array} \right]^T$$

where the bar indicates stacking. The 'memory' of the individual sources is now represented in $\mathbf{s}_{i,t}$:

$$\mathbf{s}_{i,t} = \left[\begin{array}{cccc} s_{i,t} & s_{i,t-1} & .. & s_{i,t-p+1} \end{array} \right]^T$$

The stacking procedure consists of including the last p samples of $\mathbf{s}_t$ in $\bar{\mathbf{s}}_t$ and passing the $(p-1)$ most recent of those unchanged to $\bar{\mathbf{s}}_{t+1}$ while obtaining a new $\mathbf{s}_t$ by the AR(p) recursion of equation (3). Figure 1 illustrates the principle for two AR(4) sources. The involved parameter matrices must be constrained in the following way to enforce the independency assumption:

$$\bar{\mathbf{F}} = \left[\begin{array}{cccc} \bar{\mathbf{F}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{F}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{F}}_L \end{array} \right]$$

$$\bar{\mathbf{F}}_i = \left[\begin{array}{ccccc} f_{i,1} & f_{i,2} & \cdots & f_{i,p-1} & f_{i,p} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{array} \right]$$

$$\bar{\mathbf{Q}} = \left[\begin{array}{cccc} \bar{\mathbf{Q}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{Q}}_L \end{array} \right]$$

$$(\bar{\mathbf{Q}}_i)_{jj'} = \begin{cases} q_i & j = j' = 1 \\ 0 & j \neq 1 \bigvee j' \neq 1 \end{cases}$$

Similar definitions apply to $\bar{\mathbf{\Sigma}}$ and $\bar{\mu}$. The generalization of the Kalman Filter model to represent convolutive mixing requires only a slight additional modification of the observation model, augmenting the observation matrix to a full $d_x \times p \times d_s$ matrix of filters,

$$\bar{\bar{\mathbf{A}}} = \left[\begin{array}{cccc} \mathbf{a}_{11}^T & \mathbf{a}_{12}^T & .. & \mathbf{a}_{1d_s}^T \\ \mathbf{a}_{21}^T & \mathbf{a}_{22}^T & .. & \mathbf{a}_{2d_s}^T \\ \mathbf{a}_{d_x1}^T & \mathbf{a}_{d_x2}^T & .. & \mathbf{a}_{d_xd_s}^T \end{array} \right]$$

where $\mathbf{a}_{ij} = [a_{ij,1}, a_{ij,2}, .., a_{ij,L}]^T$ is the length $L(=p)$ impulse response of the signal path between source i and sensor j. Figure 2 illustrates the the convolutive mixing matrix.

It is well-known that deconvolution cannot be performed using *stationary* second order statistics. We therefore follow Parra and Spence and *segment* the signal in windows in which the source signals can be assumed stationary. The overall system then reads

$$\begin{aligned} \bar{\mathbf{s}}_t^n &= \bar{\mathbf{F}}^n \bar{\mathbf{s}}_{t-1}^n + \bar{\mathbf{v}}_t^n \\ \mathbf{x}_t^n &= \bar{\bar{\mathbf{A}}} \bar{\mathbf{s}}_t^n + \mathbf{n}_t^n \end{aligned}$$

where n identify the segment of the observed mixture. A total of N segments are observed. For learning we will assume that during this period the mixing matrices $\bar{\bar{\mathbf{A}}}$ and the observation noise covariance, $\mathbf{R}$ are stationary.

3. LEARNING

A main benefit of having formulated the convolutive ICA problem in terms of a linear Gaussian model is that we can draw upon the extensive literature on parameter learning for such models. The likelihood is defined in abstract form for hidden variables $\mathbf{S}$ and parameters θ

$$\mathcal{L}(\theta) = \log p(\mathbf{X}|\theta) = \log \int d\mathbf{S} p(\mathbf{X}, \mathbf{S}|\theta)$$

The generic scheme for maximum likelihood learning of the parameters is the EM algorithm. The EM algorithm introduces a model posterior pdf. $\hat{p}(\cdot)$ for the hidden variables

$$\mathcal{L}(\theta) \geq \mathcal{F}(\theta, \hat{p}) \equiv \mathcal{J}(\theta, \hat{p}) - \mathcal{R}(\hat{p}) \tag{4}$$

where

$$\begin{aligned} \mathcal{J}(\theta, \hat{p}) &\equiv \int d\mathbf{S} \hat{p}(\mathbf{S}) \log p(\mathbf{X}, \mathbf{S}|\theta) \\ \mathcal{R}(\hat{p}) &\equiv \int d\mathbf{S} \hat{p}(\mathbf{S}) \log \hat{p}(\mathbf{S}) \end{aligned}$$

In the E-step we find the conditional source pdf based on the most recent parameter estimate, $\hat{p}(\mathbf{S}) = p(\mathbf{S}|\mathbf{X}, \theta)$. For linear Gaussian models we achieve $\mathcal{F}(\theta, \hat{p}) = \mathcal{L}(\theta)$. The M-step then maximize $\mathcal{J}(\theta, \hat{p})$ wrt. θ. Each combined M and E step cannot decrease $\mathcal{L}(\theta)$.

3.1 E-step

The Markov structure of the Kalman model allows an effective implementation of the E-step referred to as the *Kalman smoother*. This step involves forward-backward recursions and outputs the relevant statistics of the posterior probability $p(\bar{\mathbf{s}}_t|\mathbf{x}_{1:\tau}, \theta)$, and the log-likelihood of the parameters, $\mathcal{L}(\theta)$[1]. The posterior source mean (i.e. the posterior average conditioned on the given segment of observations) is given by

$$\hat{\bar{\mathbf{s}}}_t \equiv \langle \bar{\mathbf{s}}_t \rangle$$

for all t. The relevant second order statistics, i.e. source i autocorrelation and time-lagged autocorrelation, are:

$$\begin{aligned} \mathbf{M}_{i,t} &\equiv \langle \mathbf{s}_{i,t}(\mathbf{s}_{i,t})^T \rangle \\ &\equiv [\ \mathbf{m}_{i,1,t} \quad \mathbf{m}_{i,2,t} \quad .. \quad \mathbf{m}_{i,L,t}\ ]^T \\ \mathbf{M}^1_{i,t} &\equiv \langle \mathbf{s}_{i,t}(\mathbf{s}_{i,t-1})^T \rangle \end{aligned}$$

The block-diagonal autocorrelation matrix for $\bar{\mathbf{s}}_t$ is denoted $\bar{\mathbf{M}}_t$., It contains the individual $\mathbf{M}_{i,t}$, for $i = 1, 2, .., d_s$.

[1]For notational brevity, the segment indexing by n has been omitted in this section.

3.2 M-step

In the M-step, the first term of (4) is maximized with respect to the parameters. This involves the average of the logarithm of the data model wrt. the source posterior from the previous E-step

$$\begin{aligned} \mathcal{J}(\theta, \hat{p}) = -\frac{1}{2} \sum_{n=1}^{N} [& \sum_{i=1}^{d_s} \log\det \mathbf{\Sigma}_i^n + (\tau - 1) \sum_{i=1}^{d_s} \log q_i^n \\ & + \tau \log\det \mathbf{R} + \sum_{i=1}^{d_s} \langle (\mathbf{s}_{i,1}^n - \mu_i^n)^T (\mathbf{\Sigma}_i^n)^{-1} (\mathbf{s}_{i,1}^n - \mu_i^n) \rangle \\ & + \sum_{t=2}^{\tau} \sum_{i=1}^{d_s} \langle \frac{1}{q_i^n} (s_{i,t}^n - (\mathbf{f}_i^n)^T \mathbf{s}_{i,t-1}^n)^2 \rangle \\ & + \sum_{t=1}^{\tau} \langle (\mathbf{x}_t^n - \bar{\bar{\mathbf{A}}} \bar{\mathbf{s}}_t^n)^T \mathbf{R}^{-1} (\mathbf{x}_t^n - \bar{\bar{\mathbf{A}}} \bar{\mathbf{s}}_t^n) \rangle] \end{aligned}$$

where $\mathbf{f}_i^T = [\ f_{i,1} \quad f_{i,2} \quad .. \quad f_{i,p}\ ]$. The derivations are analogous with the formulation of the EM algorithm in [3]. The special constrained structure induced by the independency of the source signals introduces tedious but straight-forward modifications. The segment-wise update equations for the M-step are:

$$\begin{aligned} \mu_{i,\mathbf{new}} &= \hat{\mathbf{s}}_{i,1} \\ \mathbf{\Sigma}_{i,\mathbf{new}} &= \mathbf{M}_{i,1} - \mu_{i,\mathbf{new}} \mu_{i,\mathbf{new}}^T \\ \mathbf{f}_{i,\mathbf{new}}^T &= \Big[\sum_{t=2}^{\tau} (\mathbf{m}_{i,t}^1)^T \Big] \Big[\sum_{t=1}^{\tau} \mathbf{M}_{i,t-1} \Big]^{-1} \\ q_{i,\mathbf{new}} &= \frac{1}{\tau - 1} \Big[\sum_{t=2}^{\tau} m_{i,t} - \mathbf{f}_{i,\mathbf{new}}^T \mathbf{m}_{i,t}^1 \Big] \end{aligned}$$

Reconstruction of $\bar{\mu}_{\mathbf{new}}$, $\bar{\mathbf{\Sigma}}_{\mathbf{new}}$, $\bar{\mathbf{F}}_{\mathbf{new}}$ and $\bar{\mathbf{Q}}_{\mathbf{new}}$ from the above is performed according to the stacking definitions of section 2. The estimators $\bar{\bar{\mathbf{A}}}_{\mathbf{new}}$ and $\mathbf{R}_{\mathbf{new}}$ include the statistics from all observed segments:

$$\begin{aligned} \bar{\bar{\mathbf{A}}}_{\mathbf{new}} &= \Big[\sum_{n=1}^{N} \sum_{t=1}^{\tau} \mathbf{x}_{t,n} (\hat{\bar{\mathbf{s}}}_{t,n})^T \Big] \Big[\sum_{n=1}^{N} \sum_{t=1}^{\tau} \bar{\mathbf{M}}_{t,n} \Big]^{-1} \\ \mathbf{R}_{\mathbf{new}} &= \frac{1}{N\tau} \sum_{n=1}^{N} \sum_{t=1}^{\tau} \mathrm{diag}[\mathbf{x}_{t,n} \mathbf{x}_{t,n}^T - \bar{\bar{\mathbf{A}}}_{\mathbf{new}} \hat{\bar{\mathbf{s}}}_{t,n} \mathbf{x}_{t,n}^T] \end{aligned}$$

We accelerate the EM learning by a relaxation of the lower bound, which amounts to updating the parameters proportionally to an self-adjusting step-size, α, as described in [6]. We refer to the Kalman filter based blind source separation approach as 'KaBSS'.

4. EXPERIMENTS

The proposed algorithm was tested on a binaural convolutive mixture of two speech signals with additive noise in varying signal to noise ratios (SNR). A male speaker generated *both signals* that were recorded at $8kHz$. This is a strong test of the blind separation ability, since the 'spectral overlap' is maximal for a single speaker.

The noise-free mixture was obtained by convolving the source signals with the impulse responses:

$$\bar{\bar{\mathbf{A}}} = \begin{bmatrix} 1 & 0.3 & 0 & 0 & 0 & 0.8 \\ 0 & 0.8 & 0.24 & 1 & 0 & 0 \end{bmatrix}$$

Subsequently, observation noise was added in each sensor channel to construct the desired SNR. Within each experiment, the algorithm was restarted 10 times, each time estimating the parameters from 10 randomly sampled segments of length $\tau = 70$. Based on a test log-likelihood, $\mathcal{L}_{test}(\theta)$, the best estimates of $\bar{\mathbf{A}}$ and $\mathbf{R}$ were used to infer the source signals and estimate the source model ($\bar{\mathbf{F}}$ and $\bar{\mathbf{Q}}$). The model parameters were set to $p = 2$ and $L = 3$.

The separation quality was compared with the State-of-the-Art method proposed by Parra and Spence[2][5]. A signal to interference ratio (SIR): SIR $= \frac{P_{11}+P_{22}}{P_{12}+P_{21}}$ is used as comparison metric. P_{ij} is the power of the signal constituting the contribution of the ith original source to the jth source estimate. The normalized cross-correlation function was used to estimate the powers involved. The ambiguity of the source assignment was fixed prior to the SIR calculations. The results are shown in figure 3. Noise-free scenarios excepted, the new method produce better signal-to-interference values peaking at an improvement of 4dB for an SNR of 20dB. It should be noted that the present method is considerably more computational demanding than the reference method.

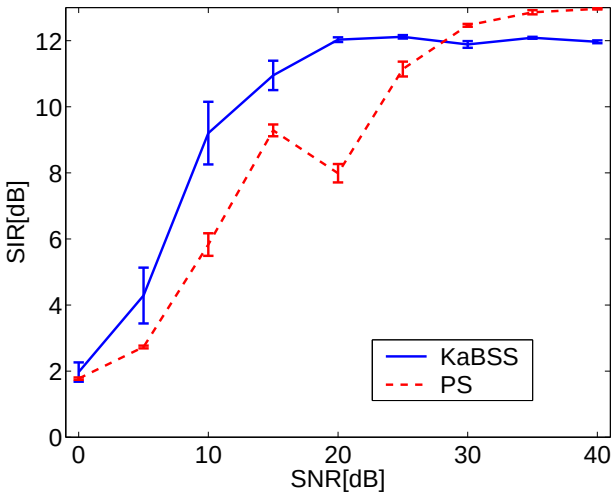

Figure 3: The separation performance for varying SNR of KaBSS and the reference method proposed by Parra and Spence (PS) [5]. The signals are two utterances by the same speaker. Two convolutive mixtures were created with variable strength additive white noise. The SIR measures the crosstalk between the two sources in the source estimates. The error bars represent the standard deviation of the mean for 10 experiments at each SNR.

5. CONCLUSION

Blind source separation of non-stationary signals has been formulated in a principled probabilistic linear Gaussian framework allowing for (exact) MAP-estimation of the sources and ML-estimation of the parameters. The derivation involved augmentation of state-space representation to model higher order AR processes and augmentation of the observation model to represent convolutive mixing. The independency constraint could be implemented exactly in the parameter estimation procedure. The source estimation and the parameter adaptation procedures are based on second-order statistics ensuring robust estimation for many classes of signals. In comparison with other current convolutive ICA models the present setup allows blind separation of noisy mixtures and it can estimate the noise characteristics. Since it is possible to compute the likelihood function on test data it is possible to both use validation sets for model order estimation as well as approximate schemes such as AIC and BIC based model order selection. A simulation study was used to validate the model in comparison with a State-of-the-Art reference method. The simulation consisted in a noisy convolutive mixture of two recordings of the *same* speaker. The simulation indicated that speech signals are described well-enough by the colored noise source model to allow separation. For the given data set, the proposed algorithm outperforms the reference method for a wide range of noise levels. However, the new method is computationally demanding. We expect that significant optimization and computational heuristics can be invoked to simplify the algorithm for real-time applications. Likewise, future work will be devoted to monitor and tune the convergence of the EM algorithm.

[2]See "http://newton.bme.columbia.edu/ lparra/publish/". The hyper-parameters of the reference method were fitted to the given data-set: $T = 1024$, $Q = 6$, $K = 7$ and $N = 5$. It should be noted that the estimated SIR is sensitive to the hyper-parameters.

REFERENCES

[1] H. Attias and C. E. Schreiner. Blind source separation and deconvolution: the dynamic component analysis algorithm. *Neural Computation*, 10(6):1373–1424, 1998.

[2] G. Doblinger. An adaptive Kalman filter for the enhancement of noisy AR signals. In *IEEE Int. Symp. on Circuits and Systems*, volume 5, pages 305–308, 1998.

[3] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto, 2 1996.

[4] T.W. Lee, A. J. Bell, and R. H. Lambert. Blind separation of delayed and convolved sources. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 758. The MIT Press, 1997.

[5] L. Parra and C. Spence. Convolutive blind separation of non-stationary sources. *IEEE Transactions Speech and Audio Processing*, pages 320–7, 5 2000.

[6] R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani. Optimization with EM and Expectation-Conjugate-Gradient. In *International Conference on Machine Learning*, volume 20, pages 672–679, 2003.

[7] S.Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

Appendix V

R. K. Olsson and L. K. Hansen, Estimating the Number of Sources in a Noisy Convolutive Mixture using BIC, in proceedings of International Conference on Independent Component Analysis and Blind Signal Separation, 618-625, 2004

Estimating the number of sources in a noisy convolutive mixture using BIC

Rasmus Kongsgaard Olsson and Lars Kai Hansen

Technical University of Denmark, Informatics and Mathematical Modelling, B321,
DK-2800 Lyngby, Denmark
email: rko@isp.imm.dtu.dk, lkh@imm.dtu.dk

Abstract. The number of source signals in a noisy convolutive mixture is determined based on the exact log-likelihoods of the candidate models. In (Olsson and Hansen, 2004), a novel probabilistic blind source separator was introduced that is based solely on the time-varying second-order statistics of the sources. The algorithm, known as 'KaBSS', employs a Gaussian linear model for the mixture, i.e. AR models for the sources, linear mixing filters and a white Gaussian noise model. Using an EM algorithm, which invokes the Kalman smoother in the E-step, all model parameters are estimated and the exact posterior probability of the sources conditioned on the observations is obtained. The log-likelihood of the parameters is computed exactly in the process, which allows for model evidence comparison assisted by the BIC approximation. This is used to determine the activity pattern of two speakers in a convolutive mixture of speech signals.

1 Introduction

We are pursuing a research program in which we aim to understand the properties of mixtures of independent source signals within a generative statistical framework. We consider *convolutive* mixtures, i.e.,

$$\mathbf{x}_t = \sum_{k=0}^{L-1} \mathbf{A}_k \mathbf{s}_{t-k} + \mathbf{n}_t, \tag{1}$$

where the elements of the source signal vector, $\mathbf{s}_t$, i.e., the d_s statistically independent source signals, are convolved with the corresponding elements of the filter matrix, $\mathbf{A}_k$. The multichannel sensor signal, $\mathbf{x}_t$, are furthermore degraded by additive Gaussian white noise.

It is well-known that separation of the source signals based on second order statistics is infeasible in general. Consider the second order statistic

$$\langle \mathbf{x}_t \mathbf{x}_{t'}^\top \rangle = \sum_{k,k'=0}^{L-1} \mathbf{A}_k \langle \mathbf{s}_{t-k} \mathbf{s}_{t'-k'}^\top \rangle \mathbf{A}_{k'}^\top + \mathbf{R},$$

where $\mathbf{R}$ is the (diagonal) noise covariance matrix. If the sources are white noise stationary, the source covariance matrix can be assumed proportional to the unit

matrix without loss of generality, and we see that the statistic is symmetric to a common rotation of all mixing matrices $\mathbf{A}_k \rightarrow \mathbf{A}_k\mathbf{U}$. This rotational invariance means that the statistic is not informative enough to identify the mixing matrix, hence, the source time series.

However, if we consider stationary sources with *known*, non-trivial, autocorrelations $\langle \mathbf{s}_t \mathbf{s}_{t'}^\top \rangle = \mathbf{C}(t-t')$, and we are given access to measurements involving multiple values of $\mathbf{C}(t-t')$, the rotational degrees of freedom are constrained and we will be able to recover the mixing matrices up to a choice of sign and scale of each source time series. Extending this argument by the observation that the mixing model (1) is invariant to filtering of a given column of the convolutive filter provided that the inverse filter is applied to corresponding source signal, we see that it is infeasible to identify the mixing matrices if these arbitrary inverse filters can be chosen to that they 'whiten' the sources.

For non-stationary sources, on the other hand, the autocorrelation functions vary through time and it is not possible to choose a single common whitening filter for each source. This means that the mixing matrices may be identifiable from multiple estimates of the second order correlation statistic (2) for non-stationary sources. Parra and Spence [1] provide analysis in terms of the number of free parameters vs. the number of linear conditions.

Also in [1], the constraining effect of source non-stationarity was exploited by simultaneously diagonalizing multiple estimates of the source power spectrum. In [2] we formulated a generative probabilistic model of this process and proved that it could estimate sources and mixing matrices in noisy mixtures. A state-space model -a Kalman filter- was specialized and augmented by a stacking procedure to model a noisy convolutive mixture of non-stationary colored noise sources, and a forward-backward EM approach was used to estimate the source statistics, mixing coefficients and the diagonal noise covariance matrix. The EM algorithm furthermore provides an exact calculation of the likelihood as it is possible to average over all possible source configurations. Other approaches based on EM schemes for source inference are [3], [4] and [5]. In [6], a non-linear state-space model is proposed.

In this presentation we elaborate on the generative model and its applications. In particular, we use the exact likelihood calculation to make inference about the dimensionality of the model, i.e. the number of sources. Choosing the incorrect model order can lead to either a too simple, biased model or a too complex model. We use the so-called Bayes Information Criterion (BIC) [7] to approximate the Bayes factor for competing hypotheses.

The model is stated in section 2, while the learning in the particular model described in section 3. Model order selection using BIC is treated in section 4. Experiments for speech mixtures are shown in section 5.

2 The model

As indicated above, the sources must be assumed non-stationary in order to uniquely retrieve the parameters and sources, since the estimation is based on

second-order statistics. In line with [1], this is obtained by *segmenting* the signals into frames, in which the wide-sense stationarity of the sources is assumed. A separate source model is assumed for each segment. The channel filters and observation noise covariance are assumed stationary across segments in the entire observed signal.

The colored noise sources are modelled by AR(p) random processes. In segment n, source i is represented by:

$$s_{i,t}^n = f_{i,1}^n s_{i,t-1}^n + f_{i,2}^n s_{i,t-2}^n + \ldots + f_{i,p}^n s_{i,t-p}^n + v_{i,t}^n \tag{2}$$

where $n \in \{1, 2, .., N\}$ and $i \in \{1, 2, .., d_s\}$. The innovation noise, $v_{i,t}$, is white Gaussian. In order to make use of well-established estimation theory, the above recursion is fitted into the framework of Gaussian linear models, for which a review is found in e.g. [8]. The Kalman filter model is an instance of this model that particularly treats continuous Gaussian linear models used widely in e.g. control and speech enhancement applications. The general Kalman filter with no control inputs is defined:

$$\begin{aligned} \mathbf{s}_t &= \mathbf{F}\mathbf{s}_{t-1} + \mathbf{v}_t \\ \mathbf{x}_t &= \mathbf{A}\mathbf{s}_t + \mathbf{n}_t \end{aligned} \tag{3}$$

where $\mathbf{v}_t$ and $\mathbf{n}_t$ are white Gaussian noise signals that drive the processes.

In order to incorporate the colored noise sources, equation (2), into the Kalman filter model, the well-known principle of *stacking* must be applied, see e.g [9]. At any time, the stacked source vector, $\bar{\mathbf{s}}_t^n$, contains the last p samples of all d_s sources:

$$\bar{\mathbf{s}}_t^n = \left[(\mathbf{s}_{1,t}^n)^\top \; (\mathbf{s}_{2,t}^n)^\top \; \ldots \; (\mathbf{s}_{d_s,t}^n)^\top \right]^\top$$

The component vectors, $\mathbf{s}_{i,t}^n$, contain the p most recent samples of the individual sources:

$$\mathbf{s}_{i,t}^n = \left[s_{i,t}^n \; s_{i,t-1}^n \; \cdots \; s_{i,t-p+1}^n \right]^\top$$

In order to maintain the statistical independency of the sources, a constrained format must be imposed on the parameters:

$$\bar{\mathbf{F}}^n = \begin{bmatrix} \bar{\mathbf{F}}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{F}}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{F}}_{d_s}^n \end{bmatrix}, \; \bar{\mathbf{F}}_i^n = \begin{bmatrix} f_{i,1}^n & f_{i,2}^n & \cdots & f_{i,p-1}^n & f_{i,p}^n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

$$\bar{\mathbf{Q}}^n = \begin{bmatrix} \bar{\mathbf{Q}}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{Q}}_{d_s}^n \end{bmatrix}, \; (\bar{\mathbf{Q}}_i^n)_{jj'} = \begin{cases} q_i^n & j = j' = 1 \\ 0 & j \neq 1 \bigvee j' \neq 1 \end{cases}$$

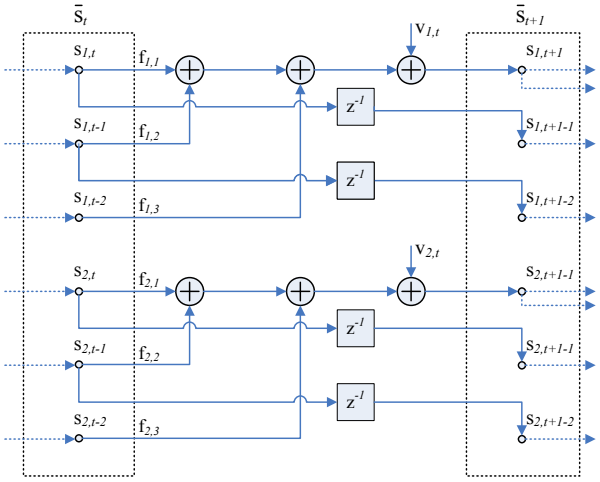

Fig. 1. The multiplication of $\bar{\mathbf{F}}$ on $\bar{\mathbf{s}}_t$ and the addition of innovation noise, $\mathbf{v}_t$, shown for an example involving two AR(3) sources. The special contrained format of $\bar{\mathbf{F}}$ simultaneously ensures the storage of past samples.

The matrix $\mathbf{A}$ of (3) is left unconstrained but its dimensions must be expanded to $d_x \times (p \times d_s)$ to reflect the stacking of the sources. Conveniently, its elements can be interpreted as the impulse responses of the channel filters of (1):

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{a}_{11}^\top & \mathbf{a}_{12}^\top & .. \, \mathbf{a}_{1d_s}^\top \\ \mathbf{a}_{21}^\top & \mathbf{a}_{22}^\top & .. \, \mathbf{a}_{2d_s}^\top \\ \mathbf{a}_{d_x1}^\top & \mathbf{a}_{d_x2}^\top & .. \, \mathbf{a}_{d_xd_s}^\top \end{bmatrix}$$

where $\mathbf{a}_{ij} = [a_{ij,1}, a_{ij,2}, .., a_{ij,L}]^\top$ is the filter between source i and sensor j. Having defined the stacked sources and the constrained parameter matrices, the total model is:

$$\begin{aligned} \bar{\mathbf{s}}_t^n &= \bar{\mathbf{F}}^n \bar{\mathbf{s}}_{t-1}^n + \bar{\mathbf{v}}_t^n \\ \mathbf{x}_t^n &= \bar{\mathbf{A}} \bar{\mathbf{s}}_t^n + \mathbf{n}_t^n \end{aligned}$$

where $\bar{\mathbf{v}}_t^n \sim (\mathbf{0}, \bar{\mathbf{Q}}^n)$ and $\mathbf{n}_t^n \sim (\mathbf{0}, \bar{\mathbf{F}}^n)$. Figures 1 and 2 illustrate the updating of the stacked source vector, $\bar{\mathbf{s}}_t$ and the effect of multiplication by $\bar{\mathbf{A}}$, respectively.

3 Learning

Having described the convolutive mixing problem in the general framework of linear Gaussian models, more specifically the Kalman filter model, optimal inference of the sources is obtained by the Kalman smoother. However,

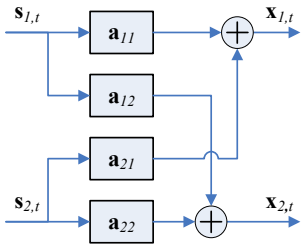

Fig. 2. The effect of the matrix multiplication $\bar{\mathbf{A}}$ on $\bar{\mathbf{s}}_t$ is shown in the system diagram. The source signals are filtered (convolved) with the impulse responses of the channel filters. Observation noise and the segment index, n, are omitted for brevity.

since the problem at hand is effectively *blind*, the parameters are estimated. Along the lines of, e.g. [8], an EM algorithm will be used for this purpose, i.e. $\mathcal{L}(\theta) \geq \mathcal{F}(\theta, \hat{p}) \equiv \mathcal{J}(\theta, \hat{p}) - \mathcal{R}(\hat{p})$, where $\mathcal{J}(\theta, \hat{p}) \equiv \int d\mathbf{S} \hat{p}(\mathbf{S}) \log p(\mathbf{X}, \mathbf{S}|\theta)$ and $\mathcal{R}(\hat{p}) \equiv \int d\mathbf{S} \hat{p}(\mathbf{S}) \log \hat{p}(\mathbf{S})$ were introduced. In accordance with standard EM theory, $\mathcal{J}(\theta, \hat{p})$ is optimized wrt. θ in the M-step. The E-step infers the model posterior, $\hat{p} = p(\mathbf{S}|\mathbf{X}, \theta)$. The combined E and M steps are guaranteed not to decrease $\mathcal{L}(\theta)$.

3.1 E-step

The forward-backward recursions which comprise the Kalman smoother is employed in the E-step to infer the source posterior, $p(\mathbf{S}|\mathbf{X}, \theta)$, i.e. the joint posterior of the sources conditioned on all observations. The relevant second-order statistics of this distribution in segment n is the posterior mean, $\hat{\bar{\mathbf{s}}}_t^n \equiv \langle \bar{\mathbf{s}}_t^n \rangle$, and autocorrelation, $\mathbf{M}_{i,t}^n \equiv \langle \mathbf{s}_{i,t}^n (\mathbf{s}_{i,t}^n)^\top \rangle \equiv [\, \mathbf{m}_{i,1,t}^n \, \mathbf{m}_{i,2,t}^n \, .. \, \mathbf{m}_{i,L,t}^n \,]^\top$, along with the time-lagged covariance, $\mathbf{M}_{i,t}^{1,n} \equiv \langle \mathbf{s}_{i,t}^n (\mathbf{s}_{i,t-1}^n)^\top \rangle \equiv [\, \mathbf{m}_{i,1,t}^{1,n} \, \mathbf{m}_{i,2,t}^{1,n} \, .. \, \mathbf{m}_{i,L,t}^{1,n} \,]^\top$. In particular, $m_{i,t}^n$ is the first element of $\mathbf{m}_{i,1,t}^n$. All averages are performed over $p(\mathbf{S}|\mathbf{X}, \theta)$. The forward recursion also yields the likelihood $\mathcal{L}(\theta)$.

3.2 M-step

The estimators are derived by straightforward optimization of $\mathcal{J}(\theta, \hat{p})$ wrt. the parameters. It is used that the data model, $p(\mathbf{X}, \mathbf{S}|\theta)$, factorizes. See, e.g., [8] for background, or [2] for details. The estimators for source i in segment n are:

$$
\begin{aligned}
\mu_{i,\mathbf{new}}^n &= \hat{\mathbf{s}}_{i,1}^n \\
\mathbf{\Sigma}_{i,\mathbf{new}}^n &= \mathbf{M}_{i,1}^n - \mu_{i,\mathbf{new}}^n (\mu_{i,\mathbf{new}}^n)^\top \\
(\mathbf{f}_{i,\mathbf{new}}^n)^\top &= \Big[\sum_{t=2}^{\tau} (\mathbf{m}_{i,t}^{1,n})^\top \Big] \Big[\sum_{t=1}^{\tau} \mathbf{M}_{i,t-1}^n \Big]^{-1} \\
q_{i,\mathbf{new}}^n &= \frac{1}{\tau - 1} \Big[\sum_{t=2}^{\tau} m_{i,t}^n - (\mathbf{f}_{i,\mathbf{new}}^n)^\top \mathbf{m}_{i,t}^{1,n} \Big]
\end{aligned}
$$

The stacked estimators, $\bar{\mu}^n_{\mathbf{new}}$, $\bar{\mathbf{\Sigma}}^n_{\mathbf{new}}$, $\bar{\mathbf{F}}^n_{\mathbf{new}}$ and $\bar{\mathbf{Q}}^n_{\mathbf{new}}$ are reconstructed from the above as defined in section 2. The constraints on the parameters cause the above estimators to differ from those of the general Kalman model, which is not the case for $\bar{\mathbf{A}}_{\mathbf{new}}$ and $\mathbf{R}_{\mathbf{new}}$:

$$\bar{\mathbf{A}}_{\mathbf{new}} = \Big[\sum_{n=1}^{N}\sum_{t=1}^{\tau}\mathbf{x}^n_t(\hat{\mathbf{s}}^n_t)^\top\Big]\Big[\sum_{n=1}^{N}\sum_{t=1}^{\tau}\bar{\mathbf{M}}^n_t\Big]^{-1}$$

$$\mathbf{R}_{\mathbf{new}} = \frac{1}{N\tau}\sum_{n=1}^{N}\sum_{t=1}^{\tau}\mathrm{diag}[\mathbf{x}^n_t(\mathbf{x}^n_t)^\top - \bar{\mathbf{A}}_{\mathbf{new}}\hat{\mathbf{s}}^n_t(\mathbf{x}^n_t)^\top]$$

4 Estimating the number of sources using BIC

In the following is described a scheme for determining d_s based on the likelihood of the parameters. A similar approach was taken in previous work, see [10]. Model control in a strictly Bayesian sense amounts to selecting the most probable hypothesis, based on the posterior probability of the model conditioned on the data:

$$p(d_s|\mathbf{X}) = \frac{p(\mathbf{X}|d_s)p(d_s)}{\sum_{d_s}p(\mathbf{X},d_s)} \tag{4}$$

In cases where all models, a priori, are to be considered equally likely, (4) reduces to $p(d_s|\mathbf{X}) \propto p(\mathbf{X}|d_s)$. The Bayes factor, $p(\mathbf{X}|d_s)$, is defined:

$$p(\mathbf{X}|d_s) = \int d\theta p(\mathbf{X}|\theta,d_s)p(\theta|d_s) \tag{5}$$

Bayes information criterion (BIC), see [7], is an approximation of (5) to be applied in cases where the marginalization of θ is intractable:

$$p(\mathbf{X}|d_s) \approx p(\mathbf{X}|\theta_{ML},d_s)\tau^{-\frac{|\theta|}{2}} \tag{6}$$

The underlying assumptions are that (5) can be evaluated by Laplace integration, i.e. $\log p(\mathbf{X}|\theta,d_s)$ is well approximated by a quadratic function for large amounts of data ($\tau \to \infty$), and that the parameter prior $p(\theta|d_s)$ can be assumed constant under the integral.

5 Experiments

In order to demonstrate the applicability of the model control setup, a convolutive mixture of speech signals was generated and added with observation noise. The four models/hypotheses that we investigate in each time frame are that only one of two speakers are active, **1** and **2**, respectively, that both of them are active, **1+2**, or that none of them are active, **0**.

Recordings of male speech[1], which were also used in [11], were filtered through the $(2 \times 2 = 4)$ known channel filters:

$$\bar{\mathbf{A}} = \begin{bmatrix} 1.00\ 0.35\ -0.20 & 0.00\ 0.00, & 0.00\ 0.00\ -0.50\ -0.30\ 0.20 \\ 0.00\ 0.00 & 0.70\ -0.20\ 0.15, & 1.30\ 0.60 \quad 0.30 \quad 0.00\ 0.00 \end{bmatrix}$$

Observation noise was added to simulate SNR=15dB in the two sensor signals. KaBSS was then invoked in order to separate the signals and estimate $\bar{\mathbf{A}}$ and $\mathbf{R}$, as shown in [2]. The signals were segmented into frames of $\tau = 160$ samples. The obtained estimates of $\bar{\mathbf{A}}$ and $\mathbf{R}$ were treated as known true parameters in the following. In each segment and for each model-configuration, KaBSS was separately reinvoked to estimate the source model parameters, $\bar{\mathbf{F}}^n$, $\bar{\mathbf{Q}}^n$, and obtain the log-likelihood, $\mathcal{L}(\theta)$, of the various models. The four resulting $\mathcal{L}(\theta)$'s were then processed in the BIC model control scheme described in section 4. The number of samples in (6) were set to τ although the sensor signals are not i.i.d. This approximation is, however, acceptable due to the noisy character of speech. Figure 3 displays the source signals, the mixtures and the most likely hypothesis in each time frame. Convincingly, the MAP speech activity detector selects the correct model.

6 Conclusion

An EM algorithm, 'KaBSS', which builds on probabilistic inference in a generative linear convolutive mixture model with Gaussian sources was introduced in [2]. This contribution expands the model and its utility by showing that the exact computation of the log-likelihood, which is readily available as an output of the forward-backward recursion, can be exploited in a BIC-based model selection scheme. The result is an exploratory tool capable of determining the correct number of sources in a convolutive mixture. In particular, it was shown that the activity pattern of two speech sources in a convolutive mixture can be well estimated. Potential applications include the ability to select the correct model in speech enhancement and communication algorithms, hopefully resulting in more robust estimation.

References

1. Parra, L., Spence C., Convolutive blind separation of non-stationary sources. IEEE Transactions, Speech and Audio Processing (5), 320-7, 2000.
2. Olsson, R. K., Hansen L. K., Probabilistic blind deconvolution of non-stationary source. Proc. EUSIPCO, 2004, *submitted*.
3. Moulines E., Cardoso J. F., Gassiat E., Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models, Proc. ICASSP (5), 3617-20, 1997.
4. Attias H., New EM algorithms for source separation and deconvolution with a microphone array. Proc. ICASSP (5), 297-300, 2003.

[1] Available at http://www.ipds.uni-kiel.de/pub_exx/bp1999_1/Proto.html.

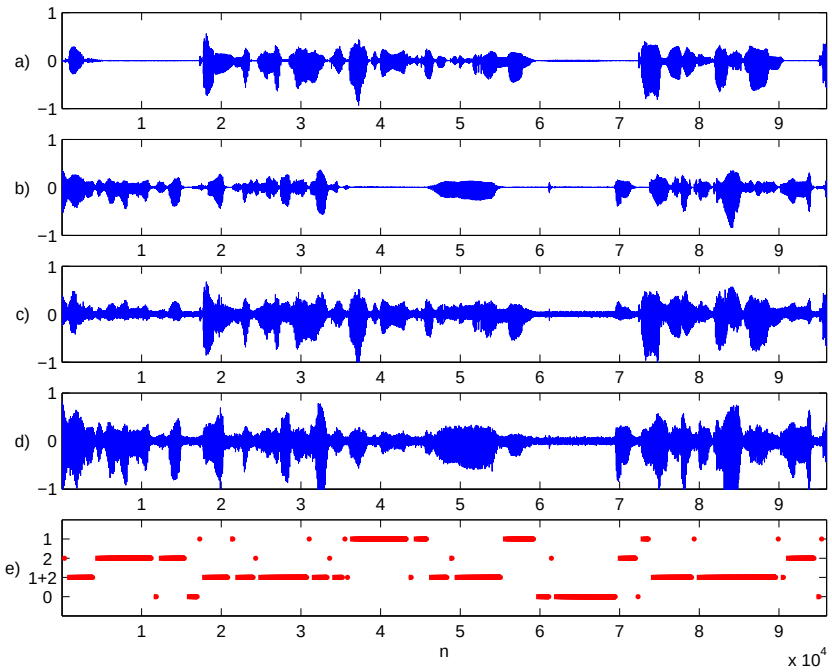

Fig. 3. From top to bottom, **a & b**) the original speech signals, **c & d**) the noisy mixtures and **e**) the most likely model in each segment. The four models are, **1**: first speaker exclusively active, **2**: second speaker exclusively active, **1+2**: both speakers simultaneously active and **0**: no speaker activity. A segment of 6 seconds of speech, sampled at $F_s = 16$kHz, is shown.

5. Todorovic-Zarkula S., Todorovic B., Stankovic M., Moraga C., Blind separation and deconvolution of nonstationary signals using extended Kalman filter. South-Eastern European workshop on comp. intelligence and IT, 2003.
6. Valpola H., Karhunen J, An unsupervised ensemble learning method for nonlinear dynamic state-space models. Neural Computation 14 (11), MIT Press, 2647-2692, 2002.
7. Schwartz G., Estimating the dimension of a model. Annals of Statistics (6), 461-464, 1978.
8. Roweis S., Ghahramani Z., Spence C., A unifying review of linear Gaussian models. Neural Computation (11), 305-345, 1999.
9. Doblinger G., An adaptive Kalman filter for the enhancement of noisy AR signals. IEEE Int. Symp. on Circuits and Systems (5), 305-308, 1998.
10. Højen-Sørensen P. A. d. F. R., Winther O., Hansen L. K., Analysis of functional neuroimages using ICA with adaptive binary sources. Neurocomputing (49), 213-225, 2002.
11. Peters B., Prototypische Intonationsmuster in deutscher Lese- und Spontansprache. AIPUK (34), 1-177, 1999.

Appendix VI

R. K. Olsson and L. K. Hansen, A Harmonic Excitation State-Space Approach to Blind Separation of Speech, in Advances in Neural Information Processing Systems, 17, eds. L. K. Saul, Y. Weiss and L. Bottou, MIT Press, 993-1000, 2005

A harmonic excitation state-space approach to blind separation of speech

Rasmus Kongsgaard Olsson and Lars Kai Hansen
Informatics and Mathematical Modelling
Technical University of Denmark, 2800 Lyngby, Denmark
rko,lkh@imm.dtu.dk

Abstract

We discuss an identification framework for noisy speech mixtures. A block-based generative model is formulated that explicitly incorporates the time-varying harmonic plus noise (H+N) model for a number of latent sources observed through noisy convolutive mixtures. All parameters including the pitches of the source signals, the amplitudes and phases of the sources, the mixing filters and the noise statistics are estimated by maximum likelihood, using an EM-algorithm. Exact averaging over the hidden sources is obtained using the Kalman smoother. We show that pitch estimation and source separation can be performed simultaneously. The pitch estimates are compared to laryngograph (EGG) measurements. Artificial and real room mixtures are used to demonstrate the viability of the approach. Intelligible speech signals are re-synthesized from the estimated H+N models.

1 Introduction

Our aim is to understand the properties of mixtures of speech signals within a generative statistical framework. We consider *convolutive* mixtures, i.e.,

$$\mathbf{x}_t = \sum_{k=0}^{L-1} \mathbf{A}_k \mathbf{s}_{t-k} + \mathbf{n}_t, \tag{1}$$

where the elements of the source signal vector, $\mathbf{s}_t$, i.e., the d_s statistically independent source signals, are convolved with the corresponding elements of the filter matrix, $\mathbf{A}_k$. The multichannel sensor signal, $\mathbf{x}_t$, is furthermore degraded by additive Gaussian white noise.

It is well-known that separation of the source signals based on second order statistics is infeasible in general. Consider the second order statistic

$$\langle \mathbf{x}_t \mathbf{x}_{t'}^\top \rangle = \sum_{k,k'=0}^{L-1} \mathbf{A}_k \langle \mathbf{s}_{t-k} \mathbf{s}_{t'-k'}^\top \rangle \mathbf{A}_{k'}^\top + \mathbf{R}, \tag{2}$$

where $\mathbf{R}$ is the (diagonal) noise covariance matrix. If the sources can be assumed stationary white noise, the source covariance matrix can be assumed proportional to the unit matrix

without loss of generality, and we see that the statistic is symmetric to a common rotation of all mixing matrices $\mathbf{A}_k \rightarrow \mathbf{A}_k\mathbf{U}$. This rotational invariance means that the acquired statistic is not informative enough to identify the mixing matrix, hence, the source time series.

However, if we consider stationary sources with *known*, non-trivial, autocorrelations $\langle \mathbf{s}_t \mathbf{s}_{t'}^\top \rangle = \mathbf{G}(t-t')$, and we are given access to measurements involving multiple values of $\mathbf{G}(t-t')$, the rotational degrees of freedom are constrained and we will be able to recover the mixing matrices up to a choice of sign and scale of each source time series. Extending this argument by the observation that the mixing model (1) is invariant to filtering of a given column of the convolutive filter provided that the inverse filter is applied to corresponding source signal, we see that it is infeasible to identify the mixing matrices if these arbitrary inverse filters can be chosen to that they are allowed to 'whiten' the sources, see also [1].

For non-stationary sources, on the other hand, the autocorrelation functions vary through time and it is not possible to choose a single common whitening filter for each source. This means that the mixing matrices may be identifiable from multiple estimates of the second order correlation statistic (2) for non-stationary sources. Analysis in terms of the number of free parameters vs. the number of linear conditions is provided in [1] and [2].

Also in [2], the constraining effect of source non-stationarity was exploited by the simultaneous diagonalization of multiple estimates of the source power spectrum. In [3] we formulated a generative probabilistic model of this process and proved that it could estimate sources and mixing matrices in noisy mixtures. Blind source separation based on state-space models has been studied, e.g., in [4] and [5]. The approach is especially useful for including prior knowledge about the source signals and for handling noisy mixtures. One example of considerable practical importance is the case of speech mixtures.

For speech mixtures the generative model based on white noise excitation may be improved using more realistic priors. Speech models based on *sinusoidal* excitation have been quite popular in speech modelling since [6]. This approach assumes that the speech signal is a time-varying mixture of a harmonic signal and a noise signal (H+N model). A recent application of this model for pitch estimation can be found in [7]. Also [8] and [9] exploit the harmonic structure of certain classes of signals for enhancement purposes. A related application is the BSS algorithm of [10], which uses the cross-correlation of the amplitude in different frequency. The state-space model naturally leads to maximum-likelihood estimation using the EM-algorithm, e.g. [11], [12]. The EM algorithm has been used in related models: [13] and [14].

In this work we generalize our previous work on state space models for blind source separation to include harmonic excitation and demonstrate that it is possible to perform simultaneous un-mixing and pitch tracking.

2 The model

The assumption of time variant source statistics help identify parameters that would otherwise not be unique within the model. In the following, the measured signals are *segmented* into frames, in which they are assumed stationary. The mixing filters and observation noise covariance matrix are assumed stationary across *all* frames.

The colored noise (AR) process that was used in [3] to model the sources is augmented to include a periodic excitation signal that is also time-varying. The specific choice of periodic basis function, i.e. the sinusoid, is motivated by the fact that the phase is linearizable,

facilitating one-step optimization. In frame n, source i is represented by:

$$\begin{aligned} s_{i,t}^n &= \sum_{t'=1}^{p} f_{i,t'}^n s_{i,t-t'}^n + \sum_{k=1}^{K} \alpha_{i,k}^n \sin(\omega_{0,i}^n kt + \beta_i^n) + v_{i,t}^n \\ &= \sum_{t'=1}^{p} f_{i,t'}^n s_{i,t-t'}^n + \sum_{k=1}^{K} c_{i,2k-1}^n \sin(\omega_{0,i}^n kt) + c_{i,2k}^n \cos(\omega_{0,i}^n kt) + v_{i,t}^n \end{aligned} \quad (3)$$

where $n \in \{1, 2, .., N\}$ and $i \in \{1, 2, .., d_s\}$. The innovation noise, $v_{i,t}^n$, is i.i.d Gaussian. Clearly, (3) represents a H+N model. The fundamental frequency, $\omega_{0,i}^n$, enters the estimation problem in an inherent non-linear manner.

In order to benefit from well-established estimation theory, the above recursion is fitted into the framework of Gaussian linear models, see [15]. The Kalman filter model is an instance of this model. The augmented state space is constructed by including a history of past samples for each source. Source vector i in frame n is defined: $\mathbf{s}_{i,t}^n = \begin{bmatrix} s_{i,t}^n & s_{i,t-1}^n & \dots & s_{i,t-p+1}^n \end{bmatrix}^\top$. All $\mathbf{s}_{i,t}^n$'s are stacked in the total source vector: $\bar{\mathbf{s}}_t^n = \begin{bmatrix} (\mathbf{s}_{1,t}^n)^\top & (\mathbf{s}_{2,t}^n)^\top & \dots & (\mathbf{s}_{d_s,t}^n)^\top \end{bmatrix}^\top$. The resulting state-space model is:

$$\begin{aligned} \bar{\mathbf{s}}_t^n &= \mathbf{F}^n \bar{\mathbf{s}}_{t-1}^n + \mathbf{C}^n \mathbf{u}_t^n + \bar{\mathbf{v}}_t^n \\ \mathbf{x}_t^n &= \mathbf{A} \bar{\mathbf{s}}_t^n + \mathbf{n}_t^n \end{aligned}$$

where $\bar{\mathbf{v}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\bar{\mathbf{s}}_1^n \sim \mathcal{N}(\mu^n, \mathbf{\Sigma}^n)$. The combined harmonics input vector is defined: $\mathbf{u}_t^n = \begin{bmatrix} (\mathbf{u}_{1,t}^n)^\top & (\mathbf{u}_{2,t}^n)^\top & \dots & (\mathbf{u}_{d_s,t}^n)^\top \end{bmatrix}^\top$, where the harmonics corresponding to source i in frame n are:

$$\mathbf{u}_{i,t}^n = \begin{bmatrix} \sin(\omega_{0,i}^n t) & \cos(\omega_{0,i}^n t) & \dots & \sin(K\omega_{0,i}^n t) & \cos(K\omega_{0,i}^n t) \end{bmatrix}^\top$$

It is apparent that the matrix multiplication by $\mathbf{A}$ constitutes a *convolutive* mixing of the sources, where the $d_x \times d_s$ channel filters are:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{11}^\top & \mathbf{a}_{12}^\top & .. & \mathbf{a}_{1d_s}^\top \\ \mathbf{a}_{21}^\top & \mathbf{a}_{22}^\top & .. & \mathbf{a}_{2d_s}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{d_x 1}^\top & \mathbf{a}_{d_x 2}^\top & .. & \mathbf{a}_{d_x d_s}^\top \end{bmatrix}$$

In order to implement the H+N source model, the parameter matrices are constrained as follows:

$$\mathbf{F}^n = \begin{bmatrix} \mathbf{F}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{F}_{d_s}^n \end{bmatrix}, \quad \mathbf{F}_i^n = \begin{bmatrix} f_{i,1}^n & f_{i,2}^n & \cdots & f_{i,p-1}^n & f_{i,p}^n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

$$\mathbf{Q}^n = \begin{bmatrix} \mathbf{Q}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}_{d_s}^n \end{bmatrix}, \qquad (\mathbf{Q}_i^n)_{jj'} = \begin{cases} q_i^n & j = j' = 1 \\ 0 & j \neq 1 \bigvee j' \neq 1 \end{cases}$$

$$\mathbf{C}^n = \begin{bmatrix} \mathbf{C}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_{d_s}^n \end{bmatrix}, \qquad \mathbf{C}_i^n = \begin{bmatrix} c_{i,1}^n & c_{i,2}^n & \cdots & c_{i,2K}^n \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

3 Learning

Having described the convolutive mixing problem in the general framework of linear Gaussian models, more specifically the Kalman filter model, optimal inference of the sources is obtained by the Kalman smoother. However, since the problem at hand is effectively *blind*, we also need to estimate the parameters. Along the lines of, e.g. [15], we will invoke an EM approach. The log-likelihood is bounded from below: $\mathcal{L}(\theta) \geq \mathcal{F}(\theta, \hat{p}) \equiv \mathcal{J}(\theta, \hat{p}) - \mathcal{R}(\hat{p})$, with the definitions $\mathcal{J}(\theta, \hat{p}) \equiv \int d\mathbf{S} \hat{p}(\mathbf{S}) \log p(\mathbf{X}, \mathbf{S}|\theta)$ and $\mathcal{R}(\hat{p}) \equiv \int d\mathbf{S} \hat{p}(\mathbf{S}) \log \hat{p}(\mathbf{S})$. In accordance with standard EM theory, $\mathcal{J}(\theta, \hat{p})$ is optimized wrt. θ in the M-step. The E-step infers the relevant moments of the marginal posterior, $\hat{p} = p(\mathbf{S}|\mathbf{X}, \theta)$. For the Gaussian model the means are also source MAP estimates. The combined E and M steps are guaranteed not to decrease $\mathcal{L}(\theta)$.

3.1 E-step

The forward-backward recursions which comprise the Kalman smoother are employed in the E-step to infer moments of the source posterior, $p(\mathbf{S}|\mathbf{X}, \theta)$, i.e. the joint posterior of the sources conditioned on all observations. The relevant second-order statistic of this distribution in segment n is the marginal posterior mean, $\hat{\bar{\mathbf{s}}}_t^n \equiv \langle \bar{\mathbf{s}}_t^n \rangle$, and autocorrelation, $\mathbf{M}_{i,t}^n \equiv \langle \mathbf{s}_{i,t}^n (\mathbf{s}_{i,t}^n)^\top \rangle \equiv [\ \mathbf{m}_{i,1,t}^n \quad \mathbf{m}_{i,2,t}^n \quad .. \quad \mathbf{m}_{i,L,t}^n\ ]^\top$, along with the marginal lag-one covariance, $\mathbf{M}_{i,t}^{1,n} \equiv \langle \mathbf{s}_{i,t}^n (\mathbf{s}_{i,t-1}^n)^\top \rangle \equiv [\ \mathbf{m}_{i,1,t}^{1,n} \quad \mathbf{m}_{i,2,t}^{1,n} \quad .. \quad \mathbf{m}_{i,L,t}^{1,n}\ ]^\top$. In particular, $m_{i,t}^n$ is the first element of $\mathbf{m}_{i,1,t}^n$. All averages are performed over $p(\mathbf{S}|\mathbf{X}, \theta)$. The forward recursion also yields the log-likelihood, $\mathcal{L}(\theta)$.

3.2 M-step

The M-step utility function, $\mathcal{J}(\theta, \hat{p})$, is defined:

$$\mathcal{J}(\theta, \hat{p}) = -\frac{1}{2} \sum_{n=1}^{N} [\sum_{i=1}^{d_s} \log\det \mathbf{\Sigma}_i^n + (\tau - 1) \sum_{i=1}^{d_s} \log q_i^n$$
$$+\tau \log\det \mathbf{R} + \sum_{i=1}^{d_s} \langle (\mathbf{s}_{i,1}^n - \mu_i^n)^T (\mathbf{\Sigma}_i^n)^{-1} (\mathbf{s}_{i,1}^n - \mu_i^n) \rangle$$
$$+ \sum_{t=2}^{\tau} \sum_{i=1}^{d_s} \langle \frac{1}{q_i^n} (s_{i,t}^n - (\mathbf{d}_i^n)^T \mathbf{z}_{i,t}^n)^2 \rangle + \sum_{t=1}^{\tau} \langle (\mathbf{x}_t^n - \mathbf{A}\bar{\mathbf{s}}_t^n)^T \mathbf{R}^{-1} (\mathbf{x}_t^n - \mathbf{A}\bar{\mathbf{s}}_t^n) \rangle]$$

where $\langle \cdot \rangle$ signifies averaging over the source posterior from the previous E-step, $p(\mathbf{S}|\mathbf{X}, \theta)$ and τ is the frame length. The linear source parameters are grouped as

$$\mathbf{d}_i^n \equiv \left[\ (\mathbf{f}_i^n)^\top \quad (\mathbf{c}_i^n)^\top\ \right]^\top \quad , \quad \mathbf{z}_i^n \equiv \left[\ (\mathbf{s}_{i,t-1}^n)^\top \quad (\mathbf{u}_{i,t}^n)^\top\ \right]^\top$$

where

$$\mathbf{f}_i^n \equiv [\ f_{i,1} \quad f_{i,2} \quad .. \quad f_{i,p}\ ]^\top \quad , \quad \mathbf{c}_i^n \equiv [\ c_{i,1} \quad c_{i,2} \quad .. \quad c_{i,p}\ ]^\top$$

Optimization of $\mathcal{J}(\theta, \hat{p})$ wrt. θ is straightforward (except for the $\omega_{0,i}^n$'s). Relatively minor changes are introduced to the estimators of e.g. [12] in order to respect the special constrained format of the parameter matrices and to allow for an external input to the model. More details on the estimators for the correlated source model are given in [3].

It is in general difficult to maximize $\mathcal{J}(\theta, \hat{p})$ wrt. to $\omega_{i,0}^n$, since several local maxima exist, e.g. at multiples of $\omega_{i,0}^n$, see e.g. [6]. This problem is addressed by narrowing the search range based on prior knowledge of the domain, e.g. that the pitch of speech lies in the range

50-400Hz. A candidate estimate for $\omega_{i,0}^n$ is obtained by computing the autocorrelation function of $s_{i,t}^n - (\mathbf{f}_i^n)^\top \mathbf{s}_{i,t-1}^n$. Grid search is performed in the vicinity of the candidate. For each point in the grid we optimize $\mathbf{d}_i^n$:

$$\mathbf{d}_{i,\mathbf{new}}^n = \left[\sum_{t=2}^{\tau} \begin{bmatrix} (\mathbf{M}_{i,t-1}^n) & \hat{\mathbf{s}}_{i,t-1}^n (\mathbf{u}_{i,t}^n)^\top \\ \mathbf{u}_{i,t}^n (\hat{\mathbf{s}}_{i,t-1}^n)^\top & \mathbf{u}_{i,t}^n (\mathbf{u}_{i,t}^n)^\top \end{bmatrix} \right]^{-1} \sum_{t=2}^{\tau} \begin{bmatrix} \mathbf{m}_{i,t,t-1}^n \\ \hat{s}_{i,t}^n \mathbf{u}_{i,t}^n \end{bmatrix} \tag{4}$$

At each step of the EM-algorithm, the parameters are normalized by enforcing $||\mathbf{A}_i|| = 1$,

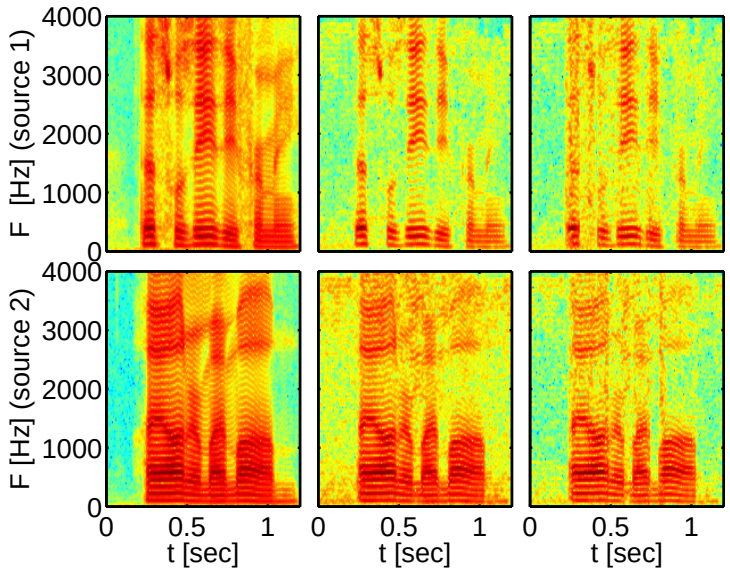

Figure 1: Amplitude spectrograms of the frequency range 0-4000Hz, from left to right: the true sources, the estimated sources and the re-synthesized source.

that is enforcing a unity norm on the filter coefficients related to source i.

4 Experiment I: BSS and pitch tracking in a noisy artificial mixture

The performance of a pitch detector can be evaluated using electro-laryngograph (EGG) recordings, which are obtained from electrodes placed on the neck, see [7]. In the following experiment, speech signals from the TIMIT [16] corpus is used for which the EGG signals were measured, kindly provided by the 'festvox' project (`http://festvox.org`).

Two male speech signals ($F_s = 16$kHz) were mixed through known mixing filters and degraded by additive white noise (SNR $\sim$20dB), constructing two observation signals. The pitches of the speech signals were overlapping. The filter coefficients (of $2 \times 2 = 4$ FIR filter impulse responses) were:

$$\mathbf{A} = \begin{bmatrix} 1.00 & 0.35 & -0.20 & 0.00 & 0.00, & 0.00 & 0.00 & -0.50 & -0.30 & 0.20 \\ 0.00 & 0.00 & 0.70 & -0.20 & 0.15, & 1.30 & 0.60 & 0.30 & 0.00 & 0.00 \end{bmatrix}$$

The signals were segmented into frames, $\tau = 320 \sim 20$ms, and the order of the AR-process was set to $p = 1$. The number of harmonics was limited to $K = 40$. The pitch grid search involved 30 re-estimations of $\mathbf{d}_i^n$. In figure 1 is shown the spectrograms of

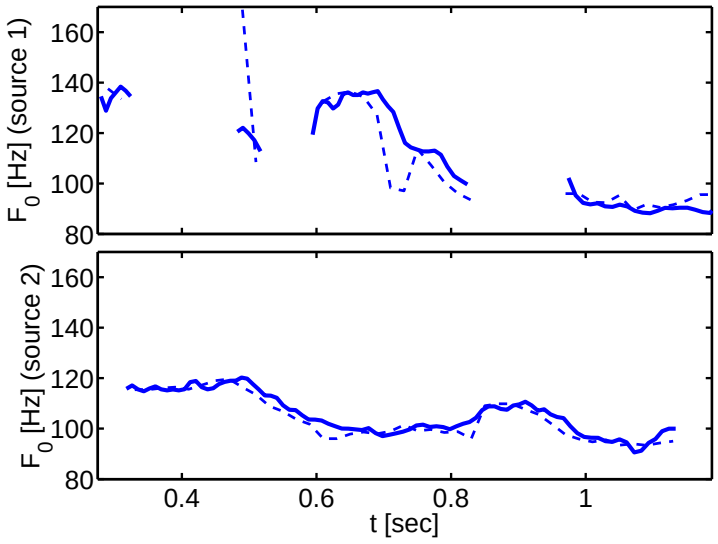

Figure 2: The estimated (dashed) and EGG-provided (solid) pitches as a function of time. The speech mixtures were artificially mixed from TIMIT utterances and white noise was added.

approximately 1 second of 1) the original sources, 2) the MAP source estimates and 3) the resynthesized sources (from the estimated model parameters). It is seen that the sources were well separated. Also, the re-synthesizations are almost indistinguishable from the source estimates. In figure 2, the estimated pitch of both speech signals are shown along with the pitch of the EGG measurements.[1] The voiced sections of the speech were manually preselected, this step is easily automated. The estimated pitches do follow the 'true' pitches as provided by the EGG. The smoothness of the estimates is further indicating the viability of the approach, as the pitch estimates are frame-local.

5 Experiment II: BSS and pitch tracking in a real mixture

The algorithm was further evaluated on real room recordings that were also used in [17].[2] Two male speakers synchronously count in English and Spanish ($F_s = 16$kHz). The mixtures were degraded with noise (SNR $\sim$20dB). The filter length, the frame length, the order of the AR-process and the number of harmonics were set to $L = 25$, $\tau = 320$, $p = 1$ and $K = 40$, respectively. Figure 3 shows the MAP source estimates and the re-synthesized sources. Features of speech such as amplitude modulation are clearly evident in estimates and re-synthesizations.[3] A listening test confirms: 1) the separation of the sources and 2) the good quality of the synthesized sources, reconfirming the applicability of the H+N model. Figure 4 displays the estimated pitches of the sources, where the voiced sections were manually preselected. Although, the 'true' pitch is unavailable in this experiment, the smoothness of the frame-local pitch-estimates is further support for the approach.

[1] The EGG data are themselves noisy measurements of the hypothesized 'truth'. Bandpass filtering was used for preprocessing.

[2] The mixtures were obtained from `http://inc2.ucsd.edu/~tewon/ica_cnl.html`.

[3] Note that the 'English' counter lowers the pitch throughout the sentence.

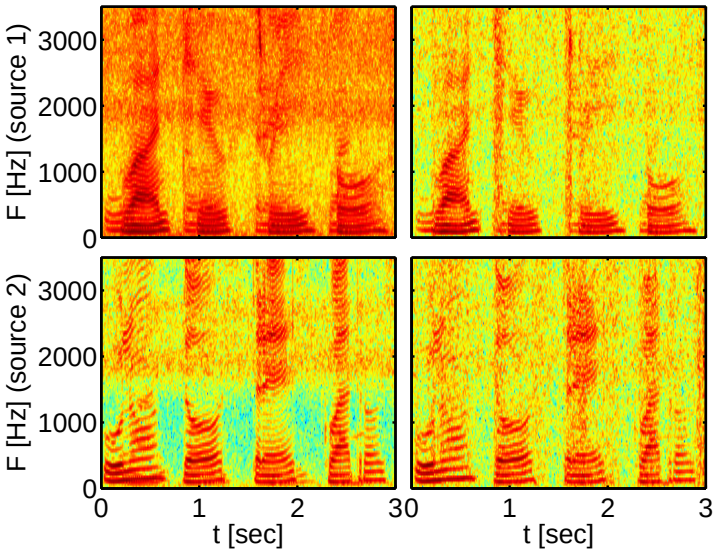

Figure 3: Spectrograms of the estimated (left) and re-synthesized sources (right) extracted from the 'one two ...' and 'uno dos ...' mixtures, source 1 and 2, respectively

6 Conclusion

It was shown that prior knowledge on speech signals and quasi-periodic signals in general can be integrated into a linear non-stationary state-space model. As a result, the simultaneous separation of the speech sources and estimation of their pitches could be achieved. It was demonstrated that the method could cope with noisy artificially mixed signals and real room mixtures. Future research concerns more realistic mixtures in terms of reverberation time and inclusion of further domain knowledge. It should be noted that the approach is computationally intensive, we are also investigating means for approximate inference and parameter estimation that would allow real time implementation.

Acknowledgement

This work is supported by the Danish 'Oticon Fonden'.

References

[1] E. Weinstein, M. Feder and A.V. Oppenheim, Multi-channel signal separation by decorrelation, IEEE Trans. on speech and audio processing, vol. 1, no. 4, pp. 405-413,1993.

[2] Parra, L., Spence C., Convolutive blind separation of non-stationary sources. IEEE Trans. on speech and audio processing, vol. 5, pp. 320-327, 2000.

[3] Olsson, R. K., Hansen L. K., Probabilistic blind deconvolution of non-stationary source. Proc. EUSIPCO, 2004, *accepted*. Olsson R. K., Hansen L. K., Estimating the number of sources in a noisy convolutive mixture using BIC. International conference on independent component analysis 2004, *accepted*. Preprints may be obtained from `http://www.imm.dtu.dk/~rko/research.htm`.

[4] Gharbi, A.B.A., Salam, F., Blind separtion of independent sources in linear dynamical media. NOLTA, Hawaii, 1993. `http://www.egr.msu.edu/bsr/papers/blind_separation/nolta93.pdf`

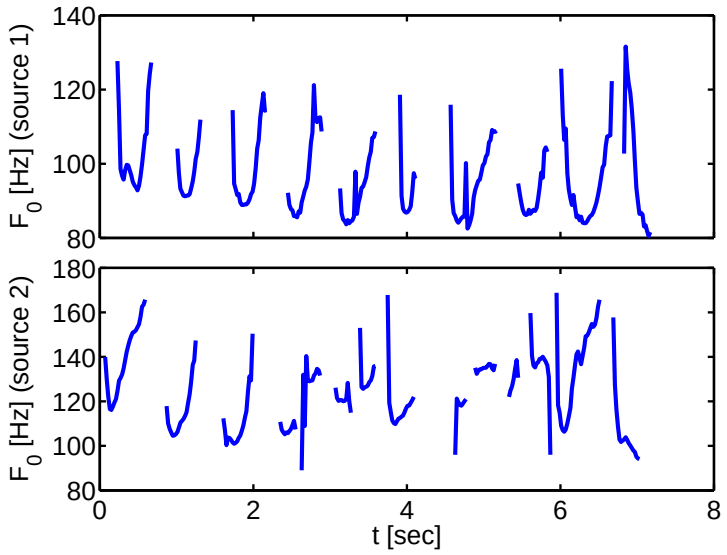

Figure 4: Pitch tracking in 'one two . . . '/'uno dos . . . ' mixtures.

[5] Zhang, L., Cichocki, A., Blind Deconvolution of dynamical systems: a state space appraoch, Journal of signal processing, vol. 4, no. 2, pp. 111-130, 2000.

[6] McAulay, R.J., Quateri. T.F., Speech analysis/synthesis based on a sinusoidal representation, IEEE Trans. on acoustics, speech and signal processing, vol. 34, no. 4, pp. 744-754, 1986.

[7] Parra, L., Jain U., Approximate Kalman filtering for the harmonic plus noise model. IEEE Workshop on applications of signal processing to audio and acoustics, pp. 75-78, 2001.

[8] Nakatani, T., Miyoshi, M., and Kinoshita, K., One microphone blind dereverberation based on quasi-periodicity of speech signals, Advances in Neural Information Processing Systems 16 (to appear), MIT Press, 2004.

[9] Hu, G. Wang, D., Monaural speech segregation based on pitch tracking and amplitude modulation, IEEE Trans. neural networks, in press, 2004.

[10] Anemüller, J., Kollmeier, B., Convolutive blind source separation of speech signals based on amplitude modulation decorrelation, Journal of the Acoustical Society of America, vol. 108, pp. 2630, 2000.

[11] A. P. Dempster, N. M. Laird, and Rubin D. B., Maximum liklihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, vol. 39, pp. 1–38, 1977.

[12] Shumway, R.H., Stoffer, D.S., An approach to time series smoothing and forecasting using the EM algorithm. Journal of time series analysis, vol. 3, pp. 253-264. 1982.

[13] Moulines E., Cardoso J. F., Gassiat E., Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models, ICASSP, vol. 5, pp. 3617-20, 1997.

[14] Cardoso, J.F., Snoussi, H. , Delabrouille, J., Patanchon, G., Blind separation of noisy Gaussian stationary sources. Application to cosmic microwave background imaging, Proc. EUSIPCO, pp 561-564, 2002.

[15] Roweis, S., Ghahramani, Z., A unifying review of linear Gaussian models. Neural Computation, vol. 11, pp. 305-345, 1999.

[16] Center for Speech Technology Research, University of Edinburgh,`http://www.cstr.ed.ac.uk/`

[17] Lee, T.-W., Bell, A.J., Orglmeister, R., Blind source separation of real world signals, Proc. IEEE international conference neural networks, pp 2129-2135, 1997.

Appendix VII

R. K. Olsson, K. B. Petersen and T. Lehn-Schiøler, State-Space Models - from the EM algorithm to a Gradient Approach, Neural Computation, 19(4), 1097-1111, 2007

LETTER **Communicated by Antti Honkela**

State-Space Models: From the EM Algorithm to a Gradient Approach

Rasmus Kongsgaard Olsson
rko@imm.dtu.dk
Kaare Brandt Petersen
kbp@epital.dk
Tue Lehn-Schiøler
tls@imm.dtu.dk
Technical University of Denmark, 2800 Kongens Lyngby, Denmark

Slow convergence is observed in the EM algorithm for linear state-space models. We propose to circumvent the problem by applying any off-the-shelf quasi-Newton-type optimizer, which operates on the gradient of the log-likelihood function. Such an algorithm is a practical alternative due to the fact that the exact gradient of the log-likelihood function can be computed by recycling components of the expectation-maximization (EM) algorithm. We demonstrate the efficiency of the proposed method in three relevant instances of the linear state-space model. In high signal-to-noise ratios, where EM is particularly prone to converge slowly, we show that gradient-based learning results in a sizable reduction of computation time.

1 Introduction

State-space models are widely applied in cases where the data are generated by some underlying dynamics. Control engineering and speech enhancement are typical examples of applications of state-space models, where the state-space has a clear physical interpretation. Black box modeling constitutes a different type of application: the state-space dynamics have no direct physical interpretation, only the generalization ability of the model matters, that is, the prediction error on unseen data.

A fairly general formulation of linear state-space models (without deterministic input) is:

$$\mathbf{s}_t = \mathbf{F}\mathbf{s}_{t-1} + \mathbf{v}_t \tag{1.1}$$

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t + \mathbf{n}_t, \tag{1.2}$$

where equations 1.1 and 1.2 describe the state and observation spaces, respectively. State and observation vectors, $\mathbf{s}_t$ and $\mathbf{x}_t$, are random processes

driven by independent and identically distributed (i.i.d.) zero-mean gaussian inputs $\mathbf{v}_t$ and $\mathbf{n}_t$ with covariance $\mathbf{Q}$ and $\mathbf{R}$, respectively.

Optimization in state-space models by maximizing the log likelihood, $\mathcal{L}(\boldsymbol{\theta})$, with respect to the parameters, $\boldsymbol{\theta} \equiv \{\mathbf{Q}, \mathbf{R}, \mathbf{F}, \mathbf{A}\}$, falls in two main categories based on either gradients (scoring) or expectation maximization (EM).

The principal approach to maximum likelihood in state-space models, and more generally in complex models, is to iteratively search the space of $\boldsymbol{\theta}$ for the local maximum of $\mathcal{L}(\boldsymbol{\theta})$ by taking steps in the direction of the gradient, $\nabla_\theta \mathcal{L}(\boldsymbol{\theta})$. A basic ascend algorithm can be improved by supplying curvature information, such as second-order derivatives or line search. Often, numerical methods are used to compute the gradient and the Hessian, due to the complexity associated with the computation of these quantities. Gupta and Mehra (1974) and Sandell and Yared (1978) give fairly complex recipes for the computation of the analytical gradient in the linear state-space model.

The EM algorithm (Dempster, Laird, & Rubin, 1977), is an important alternative to gradient-based maximum likelihood, partly due to its simplicity and convergence guarantees. It was first applied to the optimization of linear state-space models by Shumway and Stoffer (1982) and Digalakis, Rohlicek, & Ostendorf (1993). A general class of linear gaussian (state-space) models was treated in Roweis and Ghahramani (1999), in which the EM algorithm was the main engine of estimation. In the context of independent component analysis (ICA), the EM algorithm has been applied in, among others, Moulines, Cardoso, and Cassiat (1997) and Højen-Sørensen, Winther, and Hansen (2002). In Olsson and Hansen (2004, 2005), the EM algorithm was applied to the convolutive ICA problem.

A number of authors have reported the slow convergence of the EM algorithm. In Redner and Walker (1984), impractically slow convergence in two-component gaussian mixture models is documented. This critique is, however, moderated by Xu and Jordan (1996). Modifications have been suggested to speed up the basic EM algorithm (see, e.g., Lachlan & Krishnan, 1997). Jamshidian and Jennrich (1997) employ the EM update, $\tilde{\mathbf{g}}_n \equiv \boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n$, as a so-called generalized gradient in variations of the quasi-Newton algorithm. The approach taken in Meilijson (1989) differs in that the gradient of the log likelihood is derived from the expectation step (E-step), from a theorem originally shown by Fisher. Subsequently, a Newton-type step can be devised to replace the maximization step (M-step).

The main contribution of this letter is to demonstrate that specialized gradient-based optimization software can replace the EM algorithm, at little analytical cost, reusing the components of the EM algorithm itself. This procedure is termed the *easy gradient recipe*. Furthermore, empirical evidence, supporting the results in Bermond and Cardoso (1999), is presented to demonstrate that the signal-to-noise ratio (SNR) has a dramatic effect on the convergence speed of the EM algorithm. Under certain circumstances,

such as in high SNR settings, the EM algorithm fails to converge in reasonable time.

Three applications of state-space models are investigated: (1) sensor fusion for the black box modeling of speech-to-face mapping problem, (2) mean field independent component analysis (mean field ICA) for estimating a number of hidden independent sources that have been linearly and instantaneously mixed, and (3) convolutive independent component analysis for convolutive mixtures.

In section 2, an introduction to EM and the easy gradient recipe is given. More specifically, the relation between the various acceleration schemes is reviewed in section 2.3. In section 3, the models are stated, and in section 4 simulation results are presented.

2 Theory

Assume a model with observed variables $\mathbf{x}$, state-space variables $\mathbf{s}$, and parameters $\boldsymbol{\theta}$. The calculation of the log likelihood involves an integral over the state-space variables:

$$\mathcal{L}(\boldsymbol{\theta}) = \ln p(\mathbf{x}|\boldsymbol{\theta}) = \ln \int p(\mathbf{x}|\mathbf{s}, \boldsymbol{\theta}) p(\mathbf{s}|\boldsymbol{\theta}) d\mathbf{s}. \tag{2.1}$$

The marginalization in equation 2.1 is intractable for most choices of distributions, hence, direct optimization is rarely an option, even in the gaussian case. Therefore, a lower bound, B, is introduced on the log likelihood, which is valid for any choice of probability density function, $q(\mathbf{s}|\boldsymbol{\phi})$:

$$B(\boldsymbol{\theta}, \boldsymbol{\phi}) = \int q(\mathbf{s}|\boldsymbol{\phi}) \ln \frac{p(\mathbf{s}, \mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{s}|\boldsymbol{\phi})} d\mathbf{s} \leq \ln p(\mathbf{x}|\boldsymbol{\theta}). \tag{2.2}$$

At this point, the problem seems to have been made more complicated, but the lower bound B has a number of appealing properties, which makes the original task of finding the parameters easier. One important fact about B becomes clear when we rewrite it using Bayes' theorem,

$$B(\boldsymbol{\theta}, \boldsymbol{\phi}) = \ln p(\mathbf{x}|\boldsymbol{\theta}) - KL[q(\mathbf{s}|\boldsymbol{\phi})||p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})], \tag{2.3}$$

where KL denotes the Küllback-Leibler divergence between the two distributions. In the case that the variational distribution, q, is chosen to be exactly the posterior of the hidden variables, $p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$, the bound, B, is equal to the log likelihood. For this reason, one often tries to choose the variational distribution flexible enough to include the true posterior and yet simple enough to make the necessary calculations as easy as possible. However, when computational efficiency is the main priority, a simplistic

q is chosen that does not necessarily include $p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$, and the optimum of B may differ from that of $\mathcal{L}$. Examples of applications involving both types of q are described in sections 3 and 4.

The approach is to maximize B with respect to $\boldsymbol{\phi}$ in order to make the lower bound as close as possible to the log likelihood and then maximize the bound with respect to the parameters $\boldsymbol{\theta}$. This stepwise maximization can be achieved via the EM algorithm or by applying the easy gradient recipe (see section 2.2).

2.1 The EM Update. The EM algorithm, as formulated in Neal and Hinton (1998), works in a straightforward scheme that is initiated with random values and iterated until suitable convergence is reached:

E: Maximize $B(\boldsymbol{\theta}, \boldsymbol{\phi})$w.r.t. $\boldsymbol{\phi}$ keeping $\boldsymbol{\theta}$ fixed.

M: Maximize $B(\boldsymbol{\theta}, \boldsymbol{\phi})$w.r.t. $\boldsymbol{\theta}$ keeping $\boldsymbol{\phi}$ fixed.

It is guaranteed that the lower-bound function does not decrease on any combined E- and M-step. Figure 1 illustrates the EM algorithm. The convergence is often slow; for example, the curvature of the bound function, B, might be much higher than that of $\mathcal{L}$, resulting in very conservative parameter updates. As mentioned in section 1, this is particularly a problem in latent variable models with low-power additive noise. Bermond and Cardoso (1999) and Petersen and Winther (2005) demonstrate that the EM update of the parameter $\mathbf{A}$ in equation 1.2 scales with the observation noise level, $\mathbf{R}$. That is, as the signal-to-noise ratio increases, the M-step change in $\mathbf{A}$ decreases, and more iterations are required to converge.

2.2 The Easy Gradient Recipe. The key idea is to regard the bound, B, as a function of $\boldsymbol{\theta}$ only, as opposed to a function of both the parameters $\boldsymbol{\theta}$ and the variational parameters $\boldsymbol{\phi}$. As a result, the lower bound can be applied to reformulate the log likelihood,

$$\mathcal{L}(\boldsymbol{\theta}) = B(\boldsymbol{\theta}, \boldsymbol{\phi}_*), \tag{2.4}$$

where $\boldsymbol{\phi}_* = \boldsymbol{\phi}_*(\boldsymbol{\theta})$ satisfies the constraint $q(\mathbf{s}|\boldsymbol{\phi}_*) = p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$. Comparing with equation 2.3, it is easy to see that $\boldsymbol{\phi}_*$ maximizes the bound. Since $\boldsymbol{\phi}_*$ is exactly minimizing the KL divergence, the partial derivative of the bound with respect to $\boldsymbol{\phi}$ evaluated in the point $\boldsymbol{\phi}_*$ is equal to zero. Therefore, the derivative of $B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)$ is equal to the partial derivative,

$$\frac{dB(\boldsymbol{\theta}, \boldsymbol{\phi}_*)}{d\boldsymbol{\theta}} = \frac{\partial B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)}{\partial \boldsymbol{\theta}} + \frac{\partial B(\boldsymbol{\theta}, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}}\Big|_{\boldsymbol{\phi}_*} \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\phi}_*} = \frac{\partial B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)}{\partial \boldsymbol{\theta}}, \tag{2.5}$$

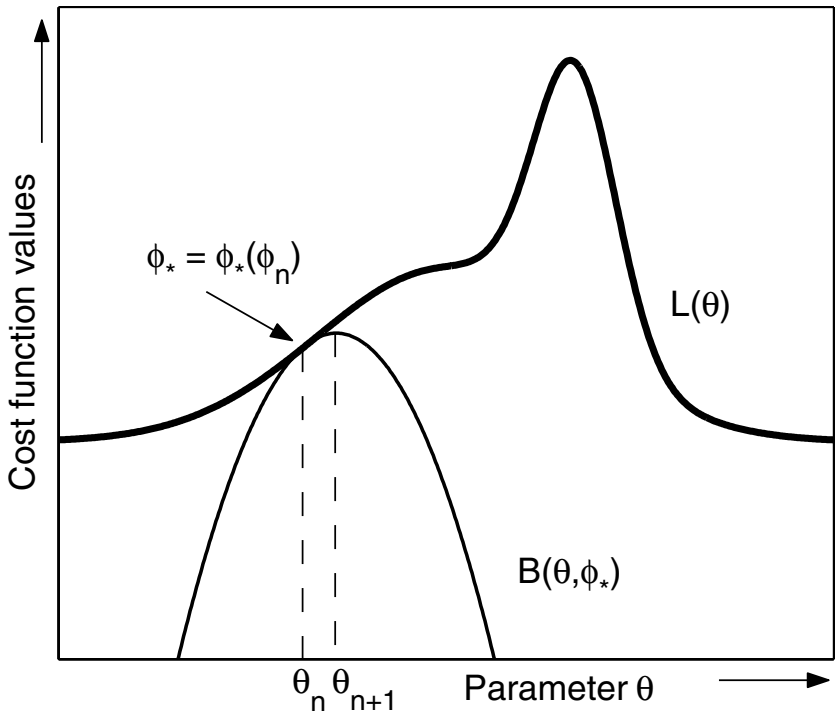

Figure 1: Schematic illustration of lower-bound optimization for a one-dimensional estimation problem, where $\boldsymbol{\theta}_n$ and $\boldsymbol{\theta}_{n+1}$ are iterates of the standard EM algorithm. The log-likelihood function, $\mathcal{L}(\boldsymbol{\theta})$, is bounded from below by the function $B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)$. The bound attains equality to $\mathcal{L}$ in $\boldsymbol{\theta}_n$ due to the choice of variational distribution: $q(\mathbf{s}|\boldsymbol{\phi}_*) = p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta}_n)$. Furthermore, in $\boldsymbol{\theta}_n$, the derivatives of the bound and the log likelihood are identical. In many situations, the curvature of $B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)$ is much higher than that of $\mathcal{L}(\boldsymbol{\theta})$, leading to small changes in the parameter, $\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n$.

and due to the choice of $\boldsymbol{\phi}_*$, the derivative of the log likelihood is the partial derivative of the bound

$$\frac{d\mathcal{L}(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \frac{dB(\boldsymbol{\theta}, \boldsymbol{\phi}_*)}{d\boldsymbol{\theta}} = \frac{\partial B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)}{\partial \boldsymbol{\theta}},$$

which can be realized by combining equations 2.4 and 2.5.

In this way, exact values and gradients of the true log likelihood can be obtained using the lower bound. This observation is not new; it is essentially the same as that used in Salakhutdinov, Roweis, and Ghahramani (2003) to construct the so-called expected conjugated gradient algorithm (ECG). The novelty of the recipe is, rather, the practical recycling of low-complexity computations carried out in connection with the EM algorithm for a much more efficient optimization using any gradient-based optimizer. This can be expressed in Matlab-style pseudocode where a function `loglikelihood` receives as argument the parameter $\boldsymbol{\theta}$ and returns $\mathcal{L}$ and its gradient $\frac{d\mathcal{L}}{d\boldsymbol{\theta}}$:

$\texttt{function}\,[\mathcal{L}, \frac{d\mathcal{L}}{d\boldsymbol{\theta}}] = \texttt{loglikelihood}(\boldsymbol{\theta})$

1. Find $\boldsymbol{\phi}^*$ such that $\frac{\partial B}{\partial \boldsymbol{\phi}}\big|_{\boldsymbol{\phi}^*} = 0$
2. Calculate $\mathcal{L} = B(\boldsymbol{\theta}, \boldsymbol{\phi}^*)$
3. Calculate $\frac{d\mathcal{L}}{d\boldsymbol{\theta}} = \frac{\partial B}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\phi}^*)$

Step 1, and to some extent step 2, are obtained by performing an E-step, while step 3 requires only little programming, which implements the gradients used to solve for the M-step. Compared to the EM algorithm, the main advantage is that the function value and gradient can be fed to any gradient-based optimizer, which in most cases substantially improves the convergence properties. In that sense, it is possible to benefit from the speed-ups of advanced gradient-based optimization. In cases when q does not contain $p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$, the easy gradient recipe will converge to generalized EM solutions.

The advantage of formulating the log likelihood using the bound function, B, depends on the task at hand. In the linear state-space model, equations 1.1 and 1.2, a naive computation of $\frac{d\mathcal{L}}{d\boldsymbol{\theta}}$ involves the somewhat complicated derivatives of the Kalman filter equations with respect to each of the parameters in $\boldsymbol{\theta}$; this is explained in, for example, Gupta and Mehra (1974). Consequently, it leads to a cost of $\dim[\boldsymbol{\theta}]$ times the cost of one Kalman filter filter sweep.[1] When using the easy gradient recipe, $\frac{d\mathcal{L}}{d\boldsymbol{\theta}}$ is derived via the gradient of the bound, $\frac{\partial B}{\partial \boldsymbol{\theta}}$ (see the appendix for a derivation example). These derivatives are often available in connection with the derivation of the M-step. In addition, the computational cost is dominated by steps 1 and 2, which require only a Kalman smoothing, scaling as two Kalman filter filter sweeps, hence constituting an important reduction of computation time as compared to the naive gradient computation. Sandell and Yared (1978) noted in in their investigation of linear state-space models that a reformulation of the problem resulted in a similar reduction of the computational costs.

2.3 Relation to Other Speed-Up Methods. In this section, a series of extensions to the EM algorithm is discussed. They have in common the utilization of conjugate gradient or quasi-Newton steps, that is, the inverse Hessian is approximated, for example, using the gradient. Step-lengthening methods, which are often simpler in terms of analytical cost, have been explored in Salakhutdinov and Roweis (2003) and Honkela, Valpola, and Karhunen (2003). They are, however, out of the scope of this presentation.

[1] The computational complexity of the Kalman filter is $\mathcal{O}[N(d_s)^3]$, where N is the data length.

Table 1: Analytical Costs Associated with a Selection of EM Speed-Ups Based on Newton-Like Updates.

Method	$\mathbf{M}$	B'	B''	$\mathcal{L}$
EM (Dempster et al., 1977)	x	-	-	-
QN1 (Jamshidian & Jennrich, 1997)	x	-	-	-
QN2 (Jamshidian & Jennrich, 1997)	x	x	-	x
CG (Jamshidian & Jennrich, 1993)	x	x	-	x
LANGE (Lange, 1995)	-	x	x	x
EGR	x	x	-	x

Notes: The x's indicate whether the quantity is required by the algorithm. It should be noted that B', required by the easy gradient recipe (EGR) as well as QN2, CG, and LANGE, often is a by-product of the derivation of $\mathbf{M}$.

In order to conveniently describe the methods, additional notation along the lines of Jamshidian and Jennrich (1997) is introduced: the combined EM operator, which maps the current estimate of the parameters, $\boldsymbol{\theta}$, to the new one, is denoted $\hat{\boldsymbol{\theta}} = \mathbf{M}(\boldsymbol{\theta})$. The change in $\boldsymbol{\theta}$ due to an EM update then is $\tilde{\mathbf{g}}(\boldsymbol{\theta}) = \mathbf{M}(\boldsymbol{\theta}) - \boldsymbol{\theta}$. In a sense, $\tilde{\mathbf{g}}(\boldsymbol{\theta})$ can be regarded a generalized gradient of an underlying cost function, which has zeros identical to those of $\mathbf{g}(\boldsymbol{\theta}) = \frac{d\mathcal{L}(\boldsymbol{\theta})}{d\boldsymbol{\theta}}$. The Newton-Raphson update can then be devised as

$$\Delta\boldsymbol{\theta} = -\mathbf{J}(\boldsymbol{\theta})^{-1}\tilde{\mathbf{g}}(\boldsymbol{\theta}),$$

where the Jacobian is defined $\mathbf{J}(\boldsymbol{\theta}) = \mathbf{M}'(\boldsymbol{\theta}) - \mathbf{I}$. A number of methods approximate $\mathbf{J}(\boldsymbol{\theta})^{-1}$ or $\mathbf{M}'(\boldsymbol{\theta})$ in various ways. For instance, the quasi-Newton method QN1 of Jamshidian and Jennrich (1997) approximates $\mathbf{J}^{-1}(\boldsymbol{\theta})$ by employing the Broyden (1965) update. The QN2 algorithm of the same publication performs quasi-Newton optimization on $\mathcal{L}(\boldsymbol{\theta})$, but its update direction can be written as an additive correction to the EM step: $\mathbf{d} = \tilde{\mathbf{g}}(\boldsymbol{\theta}) - \Psi\mathbf{g}(\boldsymbol{\theta})$. The correction term is updated through Ψ by means of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update (see, e.g., Bishop, 1995). A search is performed in the direction of $\mathbf{d}$ so as to maximize $\mathcal{L}(\boldsymbol{\theta} + \lambda\mathbf{d})$, where λ is the step length. QN2 has the disadvantage compared to QN1 that $\mathcal{L}(\boldsymbol{\theta})$ and $\mathbf{g}(\boldsymbol{\theta})$ have to be computed in addition to $\tilde{\mathbf{g}}(\boldsymbol{\theta})$. This is also the case of the conjugate gradient (CG) accelerator of Jamshidian and Jennrich (1993).

Similarly in Lange (1995), the log likelihood, $\mathcal{L}(\boldsymbol{\theta})$, is optimized directly through quasi-Newton steps, that is, in the direction of $\mathbf{d} = -\mathbf{J}_{\mathcal{L}}^{-1}(\boldsymbol{\theta})\mathbf{g}(\boldsymbol{\theta})$. However, the iterative adaptation to the Hessian, $\mathbf{J}_{\mathcal{L}}(\boldsymbol{\theta})$, involves the computation of $B''(\boldsymbol{\theta}, \boldsymbol{\phi}_*)$, which may require considerable human effort in many applications. In Table 1, the analytical cost associated with the discussed algorithms are summarized.

The easy gradient recipe lends from QN2 in that quasi-Newton optimization on $\mathcal{L}(\boldsymbol{\theta})$ is carried out. The key point here is that the specific quasi-Newton update of QN2 can be replaced by any software package of choice that performs gradient-based optimization. This has the advantage that various highly sophisticated algorithms can be tried for the particular problem at hand. Furthermore, we emphasize, as does Meilijson (1989), that the gradient can be computed from the derivatives involved in solving for the E-step. This means that the advantages of gradient-based optimization are obtained conveniently at little cost. In this letter, a quasi-Newton gradient-based optimizer has been chosen. The implementation of the BFGS algorithm is due to Nielsen (2000) and has built-in line search and trust region monitoring.

3 Models

The EM algorithm and the easy gradient recipe were applied to three models that can all be fitted into the linear state-space framework.

3.1 Kalman Filter-Based Sensor Fusion. The state-space model of equations 1.1 and 1.2 can be used to describe systems where two different types of signals are measured. The signals could be, for example, sound and images in as (Lehn-Schiøler, Hansen, & Larsen, 2005), where speech and lip movements were the observables. In this case, the observation equation 1.2, can be split into two parts,

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{x}_t^1 \\ \mathbf{x}_t^2 \end{pmatrix} = \begin{pmatrix} \mathbf{A}^1 \\ \mathbf{A}^2 \end{pmatrix} \mathbf{s}_t + \begin{pmatrix} \mathbf{n}_t^1 \\ \mathbf{n}_t^2 \end{pmatrix},$$

where $\mathbf{n}_t^1 \sim N(\mathbf{0}, \mathbf{R}^1)$ and $\mathbf{n}_t^2 \sim N(\mathbf{0}, \mathbf{R}^2)$. The innovation noise in the state-space equation 1.1, is defined as $\mathbf{v}_t \sim N(\mathbf{0}, \mathbf{Q})$. In the training phase, the parameters of the system, $\mathbf{F}$, $\mathbf{A}^1$, $\mathbf{A}^2$, $\mathbf{R}^1$,$\mathbf{R}^2$,$\mathbf{Q}$, are estimated by maximum likelihood using either EM or a gradient-based method. When the parameters have been learned, the state-space variable $\mathbf{s}$, which represents unknown hidden causes, can be deduced from one of the observations ($\mathbf{x}_1$ or $\mathbf{x}_2$), and the missing observation can be estimated by mapping from the state-space.

3.2 Mean Field ICA. In independent component analysis (ICA), one tries to separate linearly mixed sources using the assumed statistical independence of the sources. In many cases, elaborate source priors are necessary, which calls for more advanced separation techniques such as mean field ICA. The method, which was introduced in Højen-Sørensen et al. (2002), can handle complicated source priors in an efficient approximative manner.

The model in equation 1.2 is identical to an instantaneous ICA model provided that $\mathbf{F} = \mathbf{0}$ and that $p(\mathbf{v}_t)$ is reinterpreted as the (nongaussian) source prior. The basic generative model of the instantaneous ICA is

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t + \mathbf{n}_t, \tag{3.1}$$

where $\mathbf{n}_t$ is assumed i.i.d. gaussian and $\mathbf{s}_t = \mathbf{v}_t$ is assumed distributed by a factorized prior $\prod_i p(v_{it})$, which is independent in both time and dimension. The mean field ICA is only approximately compatible with the easy gradient recipe, since the variational distribution $q(\mathbf{s}|\boldsymbol{\phi})$ is not guaranteed to contain the posterior $p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$. However, acceptable solutions (generalized EM) are retrieved when q is chosen sufficiently flexible.

3.3 Convolutive ICA. Acoustic mixture scenarios are characterized by sound waves emitted by a number of sound sources propagating through the air and arriving at the sensors in delayed and attenuated versions. The instantaneous mixture model of standard ICA, equation 3.1, is clearly insufficient to describe this situation. In convolutive ICA, the signal path (delay and attenuation) is modeled by an FIR filter, that is, a convolution of the source by the impulse responses of the signal path,

$$\mathbf{x}_t = \sum_k \mathbf{C}_k \mathbf{s}_{t-k} + \mathbf{n}_t, \tag{3.2}$$

where $\mathbf{C}_k$ is the mixing filter matrix. Equation 3.2 and the source independence assumption can be fitted into the state-space formulation of equations 1.1 and 1.2 (see Olsson & Hansen, 2004, 2005), by making the following model choices: (1) noise inputs $\mathbf{v}_t$ and $\mathbf{n}_t$ are i.i.d. gaussian; (2) the state vector is augmented to contain time-lagged values, that is,

$$\bar{\mathbf{s}}_t \equiv [s_{1,t} s_{1,t-1} \ldots s_{2,t} s_{2,t-1} \ldots s_{d_s,t} s_{1,t-1} \ldots]^\top;$$

and (3) state-space parameter matrices (e.g., $\mathbf{F}$) are constrained to a special format (certain elements are fixed to 0's and 1's) in order to ensure the independence of the sources mentioned above.

4 Results

The simulations that are documented in this section serve to illustrate the well-known advantage of advanced gradient-based learning over standard EM. Before advancing to the more involved applications described above, the advantage of gradient-based methods over EM will be explored for a one-dimensional linear state-space model: an ARMA(1,1) process. In this case, $\mathbf{F}$ and $\mathbf{A}$ are scalars as well as the observation variance $\mathbf{R}$ and the

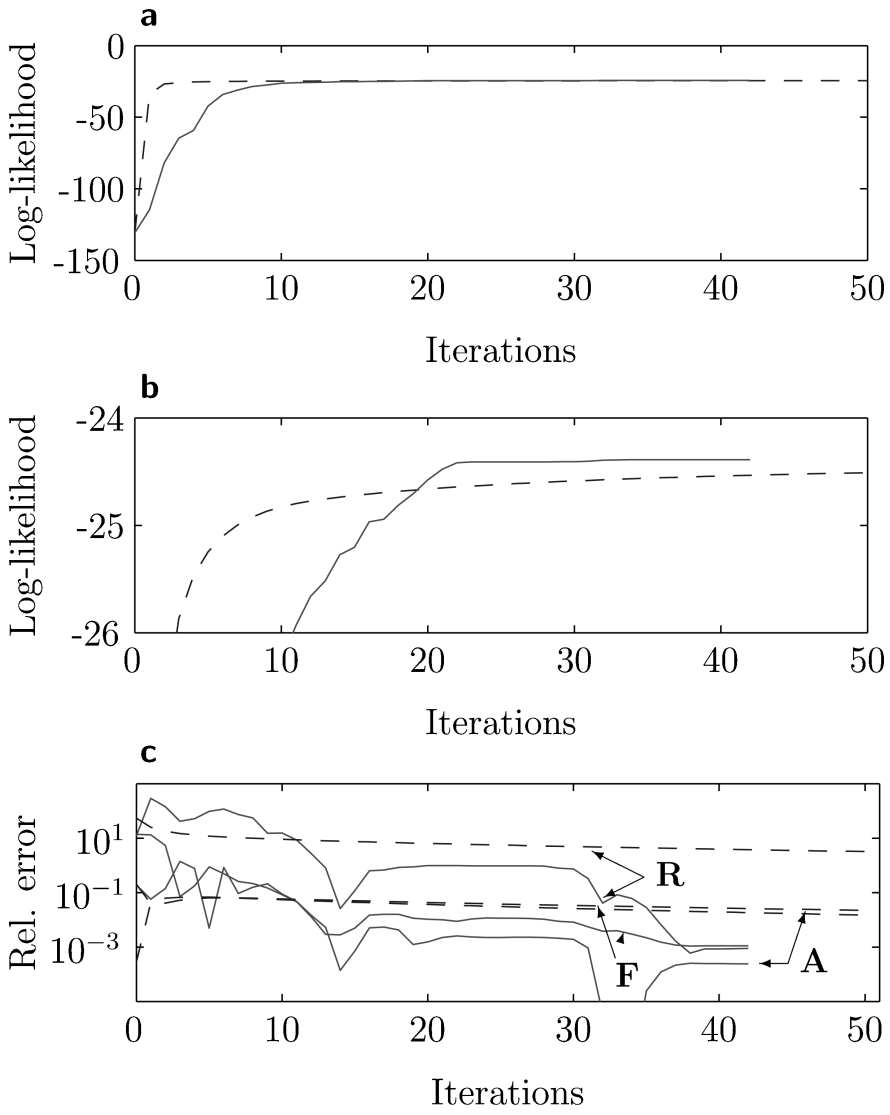

Figure 2: Convergence of EM (dashed) and a gradient-based method (solid) in the ARMA(1,1) model. (a) EM has faster initial convergence than the gradient-based method, but the final part is slow for EM. (b) Zoom-in on the log-likelihood axis. Even after 50 iterations, EM has not reached the same level as the gradient-based method. (c) Parameter estimates convergence in terms of squared relative (to the generative parameters) error.

transition variance $\mathbf{Q}$. $\mathbf{Q}$ is fixed to unity to resolve the inherent scale ambiguity of the model. As a consequence, the model has only three parameters. The BFGS optimizer mentioned in section 2 was used.

Figure 2 shows the convergence of both the EM algorithm and the gradient-based method. Initially, EM is fast; it rapidly approaches the maximum log likelihood, but slows down as it gets closer to the optimum. The large dynamic range of the log likelihood makes it difficult to ascertain the final increase in the log likelihood; hence, Figure 2b provides a close-up of the log-likelihood scale. Table 2 gives an indication of the importance of the

Table 2: Estimation in the ARMA(1,1) Model.

	Generative	Gradient	EM 50	EM ∞
Iterations	-	43	50	1800
Log likelihood	-	−24.3882	−24.5131	−24.3883
F	0.5000	0.4834	0.5626	0.4859
A	0.3000	0.2953	0.2545	0.2940
R	0.0100	0.0097	0.0282	0.0103

Notes: The convergence of EM is slow compared to the gradient-based method. Note that after 50 EM iterations, the log likelihood is relatively close to the value achieved at convergence, but the parameter values are far from the generative values.

final increase. After 50 iterations, EM has reached a log-likelihood value of −24.5131, but the parameter values are still far off. After convergence, the log likelihood has increased to −24.3883, which is still slightly worse than that obtained by the gradient-based method, but the parameters are now near the generative values. The BFGS algorithm used 43 iterations and 52 function evaluations. For large-scale state-space models, the computation time is all but dominated by the E-step computation. Hence, a function evaluation costs approximately one E-step. Similar results are obtained when comparing the learning algorithms on the Kalman filter-based sensor fusion, mean field ICA, and convolutive ICA problems.

As argued in section 2, it is demonstrated that the number of iterations required by the EM algorithm to converge in state-space type models critically depends on the SNR. Figure 3 shows the performance of the two methods on the three problems. The relevant comparison measure is computation time, which in the examples are all but dominated by the E-step for EM and function evaluations (whose cost is also dominated by an E-step) for the gradient-based optimizer. The line search of the latter may require more function evaluations per iteration, but that was most often not the case for the BFGS algorithm of choice. The plots indicate that in the low-noise case, the EM algorithm requires relatively more time to converge, whereas the gradient-based method performs equally well for all noise levels.

5 Conclusion

In applying the EM algorithm to maximum likelihood estimation in state-space models, we find, as have many before us, that it has poor convergence properties in the low noise limit. Often a value "close" to the maximum likelihood is reached in the first few iterations, while the final increase, which is crucial to the accurate estimation of the parameters, requires an excessive number of iterations.

More important, we provide a simple scheme for efficient gradient-based optimization achieved by transformation from the EM formulation; the

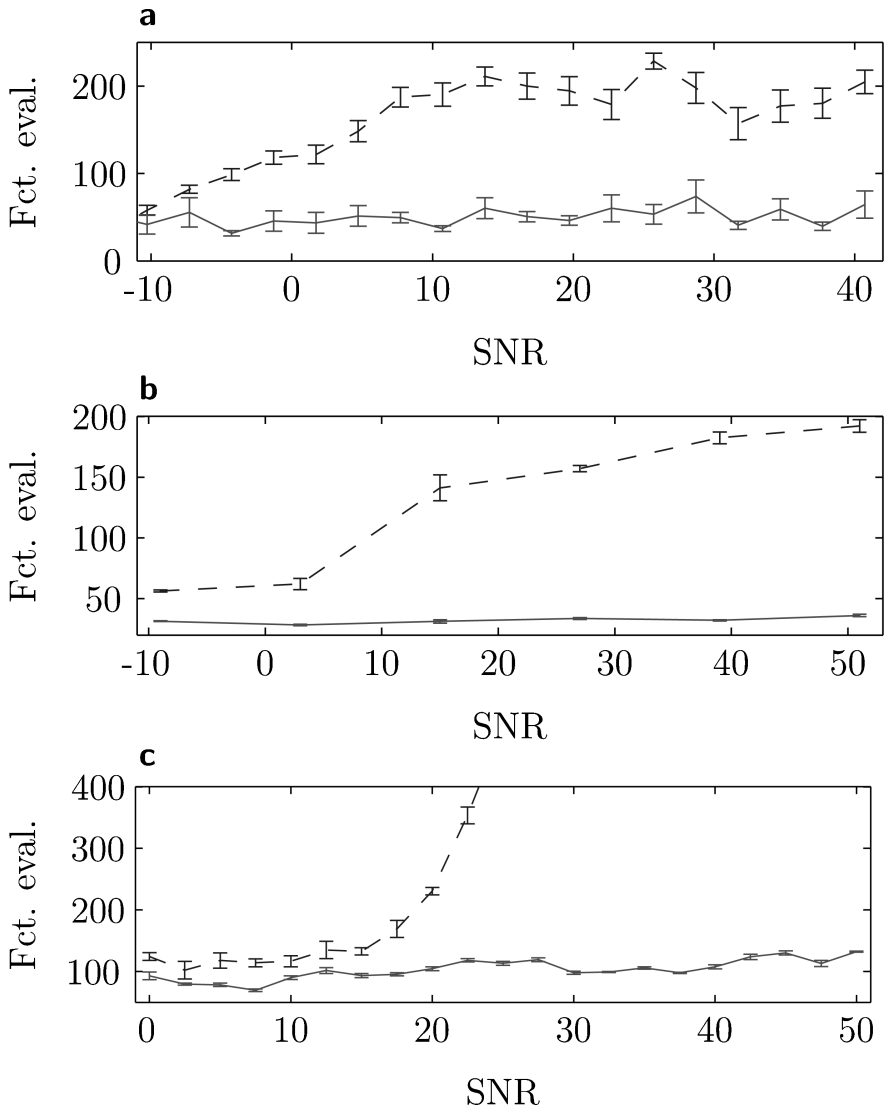

Figure 3: Number of function evaluations for EM (dashed) and gradient-based optimization (solid) to reach convergence as a function of signal-to-noise ratio for the three problems. (a) Kalman filter-based sensor fusion. (b) Mean field ICA. (c) Convolutive ICA. The level of convergence was defined as a relative change in log likehood below 10^{-5}, at which point the parameters no longer changed significantly. In the case of the EM algorithm, this sometimes occurred in plateau regions of the parameter space.

simple math and programming of the EM algorithms is preserved. Following this recipe, one can get the optimization benefits associated with any advanced gradient based-method. In this way, the tedious, problem-specific analysis of the cost-function topology can be replaced with an off-the-shelf approach. Although the analysis provided in this letter is limited to a set of linear mixture models, it is in fact applicable to any model subject to the EM algorithm, hence constituting a strong and general tool to be applied by the part of the neural community that uses the EM algorithm.

Appendix: Gradient Derivation

Here we demonstrate how to obtain a partial derivative of the constrained bound function, $B(\boldsymbol{\theta}, \boldsymbol{\phi}_*)$. The derivation lends from Shumway and Stoffer (1982). At $q(\mathbf{s}|\boldsymbol{\phi}_*) = p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$, we can reformulate equation 2.2 as

$$\begin{aligned} B(\boldsymbol{\theta}, \boldsymbol{\phi}_*) &= \left\langle \ln \frac{p(\mathbf{s}, \mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{s}|\boldsymbol{\phi}_*)} \right\rangle \\ &= \left\langle \ln p(\mathbf{s}, \mathbf{x}|\boldsymbol{\theta}) \right\rangle - \left\langle \ln q(\mathbf{s}|\boldsymbol{\phi}_*) \right\rangle, \end{aligned}$$

where the expectation $\langle \cdot \rangle$ is over the posterior $q(\mathbf{s}|\boldsymbol{\phi}_*) = p(\mathbf{s}|\mathbf{x}, \boldsymbol{\theta})$. Only the first term, $J(\boldsymbol{\theta}) = \left\langle \ln p(\mathbf{s}, \mathbf{x}|\boldsymbol{\theta}) \right\rangle$, depends on $\boldsymbol{\theta}$. The joint variable distribution factors due to the Markov property of the state-space model,

$$p(\mathbf{s}, \mathbf{x}|\theta) = p(\mathbf{s}_1|\theta) \prod_{k=2}^{N} p(\mathbf{s}_k|\mathbf{s}_{k-1}, \theta) \prod_{k=1}^{N} p(\mathbf{x}_k|\mathbf{s}_k, \theta),$$

where N is the length of the time series and the initial state, $\mathbf{s}_1$ is normally distributed with mean $\boldsymbol{\mu}_1$ and covariance $\boldsymbol{\Sigma}_1$. Using the fact that all variables are gaussian, the expected log distributions can be written out as:

$$\begin{aligned} J(\boldsymbol{\theta}) = -\frac{1}{2} \Big\{ & \dim(\mathbf{s}) \ln 2\pi + \ln |\Sigma_1| \\ & + \left\langle (\mathbf{s}_1 - \boldsymbol{\mu}_1)^\top \Sigma_1^{-1} (\mathbf{s}_1 - \boldsymbol{\mu}_0) \right\rangle \\ & + (N-1) \left[\dim(\mathbf{s}) \ln 2\pi + \ln |\mathbf{Q}| \right] \\ & + \sum_{k=2}^{N} \left\langle (\mathbf{s}_k - \mathbf{F}\mathbf{s}_{k-1})^\top \mathbf{Q}^{-1} (\mathbf{s}_k - \mathbf{F}\mathbf{s}_{k-1}) \right\rangle) \\ & + N \left[\dim(\mathbf{x}) \ln 2\pi + \ln |\mathbf{R}| \right] \\ & + \sum_{k=1}^{N} \left\langle (\mathbf{x}_k - \mathbf{A}\mathbf{s}_k)^\top \mathbf{R}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{s}_k) \right\rangle \Big\}. \end{aligned}$$

The gradient with respect to $\mathbf{A}$ is:

$$\begin{aligned} \frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{A}} &= -\frac{1}{2} \sum_{k=1}^{N} \left(2\mathbf{R}^{-1}\mathbf{A} \left\langle \mathbf{s}_k \mathbf{s}_k^\top \right\rangle - 2\mathbf{R}^{-1} \mathbf{x}_k \left\langle \mathbf{s}_k^\top \right\rangle \right) \\ &= -\mathbf{R}^{-1}\mathbf{A} \sum_{k=1}^{N} \left\langle \mathbf{s}_k \mathbf{s}_k^\top \right\rangle + \mathbf{R}^{-1} \sum_{k=1}^{N} \mathbf{x}_k \left\langle \mathbf{s}_k^\top \right\rangle). \end{aligned}$$

It is seen that the gradient depends on the marginal posterior source moments $\langle \mathbf{s}_k \mathbf{s}_k^\top \rangle$ and $\langle \mathbf{s}_k^\top \rangle$, which are provided by the Kalman smoother. The gradients with respect to $\mathbf{F}$ and $\mathbf{Q}$ furthermore require the marginal moment $\langle \mathbf{s}_k \mathbf{s}_{k-1}^\top \rangle$.

The derivation of the gradients for the covariances $\mathbf{R}$ and $\mathbf{Q}$, must respect the symmetry of these matrices. Furthermore, steps must be taken to ensure the positive definiteness following a gradient step. This can be achieved, for example, by adapting $\mathbf{Q}_0$ instead of $\mathbf{Q}$ where $\mathbf{Q} = \mathbf{Q}_0 \mathbf{Q}_0^\top$.

Acknowledgments

We thank Danish Oticon Fonden for supporting in part this research project. Inspiring discussions with Lars Kai Hansen and Ole Winther initiated this work.

References

Bermond, O., & Cardoso, J.-F. (1999). Approximate likelihood for noisy mixtures. In *Proceedings of the ICA Conference* (pp. 325–330). Aussois, France.

Bishop, C. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Math. Comput., 19*, 577–593.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistics Society, Series B, 39*, 1–38.

Digalakis, V., Rohlicek, J., & Ostendorf, M. (1993). ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing, 1*(4), 431–442.

Gupta, N. K., & Mehra, R. K. (1974). Computational aspects of maximum likelihood estimation and reduction in sensitivity calculations. *IEEE Transactions on Automatic Control, AC-19*(1), 774–783.

Højen-Sørensen, P. A., Winther, O., & Hansen, L. K. (2002). Mean field approaches to independent component analysis. *Neural Computation, 14*, 889–918.

Honkela, A., Valpola, H., & Karhunen, J. (2003). Accelerating cyclic update algorithms for parameter estimation by pattern searches. *Neural Processing Letters, 17*(2), 191–203.

Jamshidian, M., & Jennrich, R. I. (1993). Conjugate gradient acceleration of the EM algorithm. *Journal of the American Statistical Association, 88*, 221–228.

Jamshidian, M., & Jennrich, R. I. (1997). Acceleration of the EM algorithm using quasi-Newton methods. *Journal of Royal Statistical Society, B, 59*(3), 569–587.

Lachlan, G. J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: Wiley.

Lange, K. (1995). A quasi Newton acceleration of the EM algorithm. *Statistica Sinica, 5*, 1–18.

Lehn-Schiøler, T., Hansen, L. K., & Larsen, J. (2005). Mapping from speech to images using continuous state space models. In *Lecture Notes in Computer Science* (Vol. 3361, pp. 136–145). Berlin: Springer.

Meilijson, I. (1989). A fast improvement to the EM algorithm on its own terms. *Journal of Royal Statistical Society, B, 51*, 127–138.

Moulines, E., Cardoso, J., & Cassiat, E. (1997). Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Proc. ICASSP* (Vol. 5, pp. 3617–3620). Cambridge, MA: MIT Press.

Neal, R. M., & Hinton, G. E. (1998). *Learning in graphical models: A view of the EM algorithm that justifies incremental, sparse, and other variants*. Dordrecht: Kluwer.

Nielsen, H. B. (2000). *UCMINF—an algorithm for unconstrained nonlinear optimization.* (Tech. Rep. IMM-Rep-2000-19). Lungby: Technical University of Denmark.

Olsson, R. K., & Hansen, L. K. (2004). Probabilistic blind deconvolution of nonstationary sources. In *12th European Signal Processing Conference* (pp. 1697–1700). Vienna, Austria.

Olsson, R. K., & Hansen, L. K. (2005). A harmonic excitation state-space approach to blind separation of speech. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. 17, pp. 993–1000). Cambridge, MA: MIT Press.

Petersen, K. B., & Winther, O. (2005). The EM algorithm in independent component analysis. In *International Conference on Acoustics, Speech, and Signal Processing*. Piscataway, NJ: IEEE.

Redner, R. A., & Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review, 2*(26), 195–239.

Roweis, S., & Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural Computation, 11*, 305–345.

Salakhutdinov, R., & Roweis, S. (2003). Adaptive overrelaxed bound optimization methods. In *International Conference on Machine Learning* (pp. 664–671). New York: AAAI Press.

Salakhutdinov, R., Roweis, S. T., & Ghahramani, Z. (2003). Optimization with EM and expectation-conjugate-gradient. In *International Conference on Machine Learning* (Vol. 20, pp. 672–679). New York: AAAI Press.

Sandell, N. R., & Yared, K. I. (1978). *Maximum likelihood identification of state space models for linear dynamic systems.* (Tech. Rep. ESL-R-814). Cambridge, MA: Electronic Systems Laboratory, MIT.

Shumway, R., & Stoffer, D. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Anal., 3*(4), 253–264.

Xu, L., & Jordan, M. I. (1996). On convergence properties of the EM algorithm for gaussian mixtures. *Neural Computation, (8)*, 129–151.

Received February 25, 2005; accepted July 20, 2006.

Appendix VIII

R. K. Olsson and L. K. Hansen, Linear State-space Models for Blind Source Separation, Journal of Machine Learning Research, 7, 2585-2602, 2006

Journal of Machine Learning Research 7 (2006) 2585-2602 Submitted 3/06; Revised 8/06; Published 12/06

Linear State-Space Models for Blind Source Separation

Rasmus Kongsgaard Olsson RKO@IMM.DTU.DK
Lars Kai Hansen LKH@IMM.DTU.DK
Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Lyngby, Denmark

Editor: Aapo Hyvärinen

Abstract

We apply a type of generative modelling to the problem of blind source separation in which prior knowledge about the latent source signals, such as time-varying auto-correlation and quasi-periodicity, are incorporated into a linear state-space model. In simulations, we show that in terms of signal-to-error ratio, the sources are inferred more accurately as a result of the inclusion of strong prior knowledge. We explore different schemes of maximum-likelihood optimization for the purpose of learning the model parameters. The Expectation Maximization algorithm, which is often considered the standard optimization method in this context, results in slow convergence when the noise variance is small. In such scenarios, quasi-Newton optimization yields substantial improvements in a range of signal to noise ratios. We analyze the performance of the methods on convolutive mixtures of speech signals.

Keywords: blind source separation, state-space model, independent component analysis, convolutive model, EM, speech modelling

1. Introduction

We are interested in blind source separation (BSS) in which unknown source signals are estimated from noisy mixtures. Real world application of BSS techniques are found in as diverse fields as audio (Yellin and Weinstein, 1996; Parra and Spence, 2000; Anemüller and Kollmeier, 2000), brain imaging and analysis (McKeown et al., 2003), and astrophysics (Cardoso et al., 2002). While most prior work is focused on mixtures that can be characterized as instantaneous, we will here investigate causal convolutive mixtures. The mathematical definitions of these classes of mixtures are given later in this introductory section. Convolutive BSS is relevant in many signal processing applications, where the instantaneous mixture model cannot possibly capture the latent causes of the observations due to different time delays between the sources and sensors. The main problem is the lack of general models and estimation schemes; most current work is highly application specific with the majority focused on applications in separation of speech signals. In this work we will also be concerned with speech signals, however, we will formulate a generative model that may be generalizable to several other application domains.

One of the most successful approaches to convolutive BSS is based on the following assumptions: 1) The mixing process is linear and causal, 2) the source signals are statistically independent, 3) the sources can be fully characterized by their *time variant* second order statistics (Weinstein et al., 1993; Parra and Spence, 2000). The last assumption is defining for this approach. Keeping to second order statistics we simplify computations but have to pay the price of working with time-

variant statistics. It is well-known that stationary second order statistics, that is, covariances and correlation functions, are not informative enough in the convolutive mixing case.

Our research concerns statistical analysis and generalizations of this approach. We formulate a generative model based on the same statistics as the Parra-Spence model. The benefit of this generative approach is that it allows for estimation of additional noise parameters and injection of well-defined a priori information in a Bayesian sense (Olsson and Hansen, 2005). Furthermore, we propose several algorithms to learn the parameters of the proposed models.

The linear mixing model reads

$$\mathbf{x}_t = \sum_{k=0}^{L-1} \mathbf{A}_k \mathbf{s}_{t-k} + \mathbf{w}_t. \tag{1}$$

At discrete time t, the observation vector, $\mathbf{x}_t$, results from the convolution sum of the L time-lagged mixing matrices $\mathbf{A}_k$ and the source vector $\mathbf{s}_t$. The individual sources, that is, the elements of $\mathbf{s}_t$, are assumed to be statistically independent. The observations are corrupted by additive i.i.d. Gaussian noise, $\mathbf{w}_t$. BSS is concerned with estimating $\mathbf{s}_t$ from $\mathbf{x}_t$, while $\mathbf{A}_k$ is unknown. It is apparent from (1) that only filtered versions of the elements of $\mathbf{s}_t$ can be retrieved, since the inverse filtering can be applied to the unknown $\mathbf{A}_k$. As a special case of the filtering ambiguity, the *scale* and the ordering of the sources is unidentifiable. The latter is evident from the fact that various permutation applied simultaneously to the elements of $\mathbf{s}_t$ and the columns of $\mathbf{A}_t$ produce identical mixtures, $\mathbf{x}_t$.

Equation (1) collapses to an *instantaneous* mixture in the case of $L = 1$ for which a variety of Independent Component Analysis (ICA) methods are available (e.g., Comon, 1994; Bell and Sejnowski, 1995; Hyvarinen et al., 2001). As already mentioned, however, we will treat the class of convolutive mixtures, that is $L > 1$.

Convolutive Independent Component Analysis (C-ICA) is a class of BSS methods for (1) where the source estimates are produced by computing the 'unmixing' transformation that restores statistical independence. Often, an inverse linear filter (e.g., FIR) is applied to the observed mixtures. Simplistically, the separation filter is estimated by minimizing the mutual information, or 'cross' moments, of the 'separated' signals. In many cases non-Gaussian models/higher-order statistics are required, which require a relatively long data series for reliable estimation. This can be executed in the time domain (Lee et al., 1997; Dyrholm and Hansen, 2004), or in the frequency domain (e.g., Parra and Spence, 2000). The transformation to the Fourier domain reduces the matrix convolution of (1) to a matrix product. In effect, the more difficult convolutive problem is decomposed into a number of manageable instantaneous ICA problems that can be solved independently using the mentioned methods. However, frequency domain decomposition suffers from *permutation over frequency* which is a consequence of the potential different orderings of sources at different frequencies. Many authors have explored solutions to the permutation-over-frequency problem that are based on measures of spectral structure (e.g., Anemüller and Kollmeier, 2000), where amplitude correlation across frequency bands is assumed and incorporated in the algorithm.

The work presented here forges research lines that treat instantaneous ICA as a density estimation problem (Pearlmutter and Parra, 1997; Højen-Sørensen et al., 2002), with richer source priors that incorporate time-correlation, non-stationarity, periodicity and the convolutive mixture model to arrive at an C-ICA algorithm. The presented algorithm, which operates entirely in the time-domain, relies on a linear state-space model, for which estimation and exact source inference are available. The states directly represent the sources, and the transition structure can be interpreted as describing the internal time-correlation of the sources. To further increase the audio realism of the model,

Olsson and Hansen (2005) added a harmonic excitation component in the source speech model (Brandstein, 1998); this idea is further elaborated and tested here.

Algorithms for the optimization of the likelihood of the linear state-space model are devised and compared, among them the basic EM algorithm, which is used extensively in latent variable models (Moulines et al., 1997). In line with Bermond and Cardoso (1999), the EM-algorithm is shown to exhibit slow convergence in good signal to noise ratios.

It is interesting that the two 'unconventional' aspects of our generative model: the non-stationarity of the source signals and their harmonic excitation, do not change the basic quality of the state-space model, namely that exact inference of the sources and exact calculation of the log-likelihood and its gradient are still possible.

The paper is organized as follows: First we introduce the state-space representation of the convolutive mixing problem and the source models in Section 2, in Section 3 we briefly recapitulate the steps towards exact inference for the source signals, while Section 4 is devoted to a discussion of parameter learning. Sections 5 and 6 present a number of experimental illustrations of the approach on simulated and speech data respectively.

2. Model

The convolutive blind source separation problem is cast as a density estimation task in a latent variable model as was suggested in Pearlmutter and Parra (1997) for the instantaneous ICA problem

$$\mathrm{p}(\mathbf{X}|\theta) = \int \mathrm{p}(\mathbf{X}|\mathbf{S},\theta_1)\mathrm{p}(\mathbf{S}|\theta_2)d\mathbf{S}.$$

Here, the matrices $\mathbf{X}$ and $\mathbf{S}$ are constructed as the column sets of $\mathbf{x}_t$ and $\mathbf{s}_t$ for all t. The functional forms of the conditional likelihood, $\mathrm{p}(\mathbf{X}|\mathbf{S},\theta_1)$, and the joint source prior, $\mathrm{p}(\mathbf{S}|\theta_2)$, should ideally be selected to fit the realities of the separation task at hand. The distributions depend on a set of tunable parameters, $\theta \equiv \{\theta_1,\theta_2\}$, which in a blind separation setup is to be learned from the data. In the present work, $\mathrm{p}(\mathbf{X}|\mathbf{S},\theta_1)$ and $\mathrm{p}(\mathbf{S}|\theta_2)$ have been restricted to fit into a class of linear state-space models, for which effective estimation schemes exist (Roweis and Ghahramani, 1999)

$$\mathbf{s}_t = \mathbf{F}^n\mathbf{s}_{t-1} + \mathbf{C}^n\mathbf{u}_t + \mathbf{v}_t, \quad (2)$$

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t + \mathbf{w}_t. \quad (3)$$

Equations (2) and (3) describe the *state/source* and *observation* spaces, respectively. The parameters of the former are time-varying, indexed by the block index n, while the latter noisy mixing process is stationary. The randomness of the model is enabled by i.i.d. zero mean Gaussian variables, $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0},\mathbf{Q}^n)$, and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0},\mathbf{R})$ The 'input' or 'control' signal $\mathbf{u}_t \equiv \mathbf{u}_t(\psi^n)$ deterministically shifts the mean of $\mathbf{s}_t$ depending on parameters ψ^n. Various structures can be imposed on the model parameters, $\theta_1 = \{\mathbf{A},\mathbf{R}\}$ and $\theta_2 = \{\mathbf{F}^n,\mathbf{C}^n,\mathbf{Q}^n,\psi^n\}$, in order to create the desired effects. For equations (2) and (3) to pose as a generative model for the instantaneous mixture of first-order autoregressive, AR(1), sources it need only be assumed that $\mathbf{F}^n$ and $\mathbf{Q}^n$ are diagonal matrices and that $\mathbf{C}^n = \mathbf{0}$. In this case, $\mathbf{A}$ functions as the mixing matrix. In Section 2.1, we generalize to AR(p) and convolutive mixing.

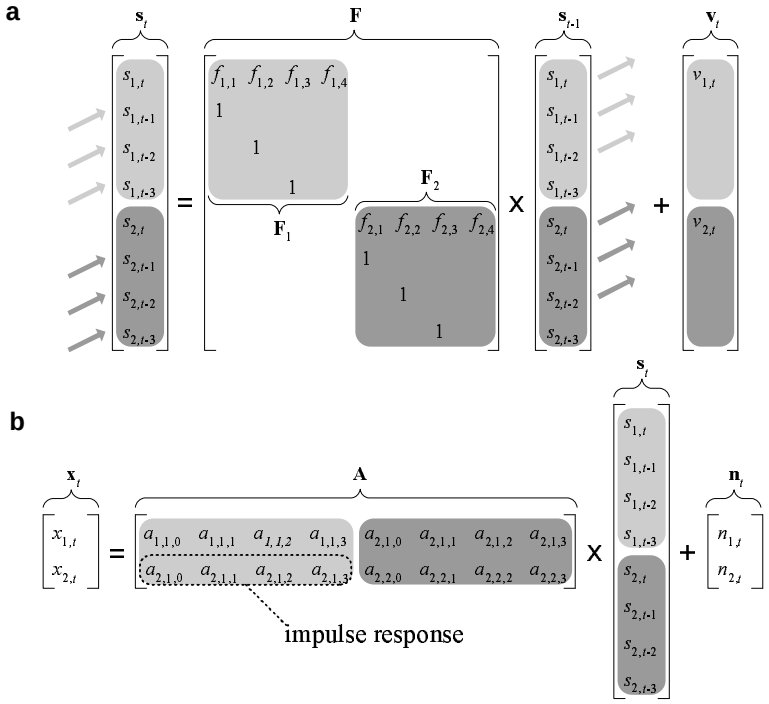

Figure 1: The dynamics of the linear state space model when it has been constrained to describe a noisy convolutive mixture of $P = 2$ autoregressive (AR) sources. This is achieved by augmenting the source vector to contain time-lagged signals. In **a** is shown the corresponding source update, when the order of the AR process is $p = 4$. In **b**, the sources are mixed through filters ($L = 4$) into $Q = 2$ noisy mixtures. Blanks signify zeros.

2.1 Auto-Regressive Source Prior

The AR(p) source prior for source *i* in frame *n* is defined as follows,

$$s_{i,t} = \sum_{k=1}^{p} f_{i,k}^{n} s_{i,t-k} + v_{i,t}$$

where $t \in \{1,2,..,T\}$, $n \in \{1,2,..,N\}$ and $i \in \{1,2,..,P\}$. The excitation noise is i.i.d. zero mean Gaussian: $v_{i,t} \sim \mathcal{N}(0, q_i^n)$. It is an important point that the convolutive mixture of AR(p) sources can be contained in the linear state-space model (2) and (3), this is illustrated in Figure 1. The enabling trick, which is standard in time series analysis, is to augment the source vector to include a time history so that it contains L time-lagged samples of all P sources

$$\mathbf{s}_t = \left[\begin{array}{cccc} (\mathbf{s}_{1,t})^\top & (\mathbf{s}_{2,t})^\top & \ldots & (\mathbf{s}_{P,t})^\top \end{array} \right]^\top$$

where the *i*'th source is represented as

$$\mathbf{s}_{i,t} = \left[\begin{array}{cccc} s_{i,t} & s_{i,t-1} & \ldots & s_{i,t-L+1} \end{array} \right]^\top.$$

Furthermore, constraints are enforced on the matrices of θ

$$\mathbf{F}^n = \begin{bmatrix} \mathbf{F}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{F}_P^n \end{bmatrix}, \qquad \mathbf{Q}^n = \begin{bmatrix} \mathbf{Q}_1^n & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2^n & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}_P^n \end{bmatrix},$$

$$\mathbf{F}_i^n = \begin{bmatrix} f_{i,1}^n & f_{i,2}^n & \cdots & f_{i,p-1}^n & f_{i,p}^n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad (\mathbf{Q}_i^n)_{jj'} = \begin{cases} (q_i^2)^n & j = j' = 1 \\ 0 & j \neq 1 \bigvee j' \neq 1 \end{cases},$$

$$\mathbf{C}^n = \mathbf{0},$$

where $\mathbf{F}_i^n$ was defined for $p = L$. In the interest of the simplicity of the presentation, it is assumed that $\mathbf{F}_i^n$ has L row and columns. We furthermore assume that $p \leq L$; in the case of $p < L$, zeros replace the affected (rightmost) coefficients. Hence, the dimensionality of $\mathbf{A}$ is $Q \times (p \times P)$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{11}^\top & \mathbf{a}_{12}^\top & .. & \mathbf{a}_{1P}^\top \\ \mathbf{a}_{21}^\top & \mathbf{a}_{22}^\top & .. & \mathbf{a}_{2P}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{Q1}^\top & \mathbf{a}_{Q2}^\top & .. & \mathbf{a}_{QP}^\top \end{bmatrix}$$

where $\mathbf{a}_{ij} = [a_{ij,1}, a_{ij,2}, .., a_{ij,L}]^\top$ can be interpreted as the impulse response of the channel filter between source i and sensor j. Overall, the model can described can be described as the generative, time-domain equivalent of Parra and Spence (2000).

2.2 Harmonic Source Prior

Many classes of audio signals, such as voiced speech and musical instruments, are approximately piece-wise periodic. By the Fourier theorem, such sequences can be represented well by a harmonic series. In order to account for colored noise residuals and noisy signals in general, a harmonic and noise (HN) model is suggested (McAulay and Quateri, 1986). The below formulation is used

$$s_{i,t} = \sum_{t'=1}^{p} f_{i,t'}^n s_{i,t-t'} + \sum_{k=1}^{K} \left[c_{i,2k-1}^n \sin(\omega_{0,i}^n t) + c_{i,2k}^n \cos(\omega_{0,i}^n t) \right] + v_{i,t}$$

where $\omega_{0,i}^n$' is the fundamental frequency of source i in frame n and the Fourier coefficients are contained in $c_{i,2k-1}^n$ and $c_{i,2k}^n$. The harmonic model is represented in the state space model (2) & (3)

through the definitions

$$\begin{aligned}
\mathbf{C}^n &= \begin{bmatrix} (\mathbf{c}_1^n)^\top & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\mathbf{c}_2^n)^\top & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\mathbf{c}_P^n)^\top \end{bmatrix}, \\
\mathbf{c}_i^n &= \begin{bmatrix} c_{i,1}^n & c_{i,2}^n & \cdots & c_{i,2K}^n \end{bmatrix}^\top, \\
\mathbf{u}_t^n &= \begin{bmatrix} (\mathbf{u}_{1,t}^n)^\top & (\mathbf{u}_{2,t}^n)^\top & \ldots & (\mathbf{u}_{P,t}^n)^\top \end{bmatrix}^\top,
\end{aligned}$$

where the k'th harmonics of source i in frame n are defined as $(\mathbf{u}_{i,t}^n)_{2k-1} = \sin(k\omega_{0,i}^n t)$ and $(\mathbf{u}_{i,t}^n)_{2k} = \cos(k\omega_{0,i}^n t)$, implying the following parameter set for the source mean: $\psi^n = \begin{bmatrix} \omega_{0,1}^n & \omega_{0,2}^n & \ldots & \omega_{0,P}^n \end{bmatrix}$. Other parametric mean functions could, of course, be used, for example, a more advanced speech model.

3. Source Inference

In a maximum a posteriori sense, the sources, $\mathbf{s}_t$, can be optimally reconstructed using the Kalman filter/smoother (Kalman and Bucy, 1960; Rauch et al., 1965). This is based on the assumption that the parameters θ are known, either a priori or have been estimated as described in Section 4. While the filter computes the time-marginal moments of the source posterior conditioned on past and present samples, that is, $\langle \mathbf{s}_t \rangle_{p(\mathbf{S}|\mathbf{x}_{1:t},\theta)}$ and $\langle \mathbf{s}_t \mathbf{s}_t^\top \rangle_{p(\mathbf{S}|\mathbf{x}_{1:t},\theta)}$, the smoother conditions on samples from the entire block: $\langle \mathbf{s}_t \rangle_{p(\mathbf{S}|\mathbf{x}_{1:T},\theta)}$ and $\langle \mathbf{s}_t \mathbf{s}_t^\top \rangle_{p(\mathbf{S}|\mathbf{x}_{1:T},\theta)}$. For the Kalman filter/smoother to compute MAP estimates, it is a precondition due that the model is linear and Gaussian. The computational complexity is $O(TL^3)$ due to a matrix inversion occurring in the recursive update. Note that the forward recursion also yields the exact log-likelihood of the parameters given the observations, $\mathcal{L}(\theta)$. A thorough review of linear state-space modelling, estimation and inference from a machine learning point of view can be found in Roweis and Ghahramani (1999).

4. Learning

The task of learning the parameters of the state-space model from data is approached by maximum-likelihood estimation, that is, the log-likelihood function, $\mathcal{L}(\theta)$, is optimized with respect to the parameters, θ. The log-likelihood is defined as a marginalization over the hidden sources

$$\mathcal{L}(\theta) = \log \mathrm{p}(\mathbf{X}|\theta) = \log \int \mathrm{p}(\mathbf{X}, \mathbf{S}|\theta) d\mathbf{S}.$$

A closed-form solution, $\theta = \arg\max_{\theta'} \mathcal{L}(\theta')$, is not available, hence iterative algorithms that optimize $\mathcal{L}(\theta)$ are employed. In the following sections three such algorithms are presented.

4.1 Expectation Maximization Algorithm

Expectation Maximization (EM) (Dempster et al., 1977), has been applied to latent variable models in, for example, Shumway and Stoffer (1982) and Roweis and Ghahramani (1999). In essence, EM

is iterative optimization of a lower bound decomposition of the log-likelihood

$$\mathcal{L}(\theta) \quad \geq \quad \mathcal{F}(\theta,\hat{p}) = \mathcal{J}(\theta,\hat{p}) - \mathcal{R}(\hat{p}) \tag{4}$$

where $\hat{p}(\mathbf{S})$ is any normalized distribution and the following definitions apply

$$\begin{aligned} \mathcal{J}(\theta,\hat{p}) &= \int \hat{p}(\mathbf{S}) \log p(\mathbf{X},\mathbf{S}|\theta) d\mathbf{S}, \\ \mathcal{R}(\hat{p}) &= \int \hat{p}(\mathbf{S}) \log \hat{p}(\mathbf{S}) d\mathbf{S}. \end{aligned}$$

Jensen's inequality leads directly to (4). The algorithm alternates between performing Expectation (E) and Maximization (M) steps, guaranteeing that $\mathcal{L}(\theta)$ does not decrease following an update. On the E-step, the Kalman smoother is used to compute the marginal moments from the source posterior, $\hat{p} = p(\mathbf{S}|\mathbf{X},\theta)$, see Section 3. The M-step amounts to optimization of $\mathcal{J}(\theta,\hat{p})$ with respect to θ (since this is the only $\mathcal{F}(\theta,\hat{p})$ term which depends on θ. Due to the choice of a linear Gaussian model, closed-form estimators are available for the M-step (see appendix A for derivations).

In order to improve on the convergence speed of the basic EM algorithm, the search vector devised by the M-step update is premultiplied by an adaptive step-size η. A simple exponentially increase of η from 1 was used until a decrease in $\mathcal{L}(\theta)$ was observed at which point η was reset to 1. This speed-up scheme was applied successfully in Salakhutdinov and Roweis (2003). Below follow the M-step estimators for the AR and HN models. All expectations $\langle\cdot\rangle$ are over the source posterior, $p(\mathbf{S}|\mathbf{X},\theta)$:

4.1.1 AUTOREGRESSIVE MODEL

For source i in block n:

$$\begin{aligned} \mathbf{f}^n_{i,\mathtt{new}} &= \Big[\sum_{t=2+t_0(n)}^{T+t_0(n)} \langle \mathbf{s}_{i,t-1}\mathbf{s}_{i,t-1}^\top \rangle\Big]^{-\top} \Big[\sum_{t=2+t_0(n)}^{T+t_0(n)} \langle s_{i,t}\mathbf{s}_{i,t-1}\rangle\Big], \\ q^n_{i,\mathtt{new}} &= \frac{1}{T-1}\sum_{t=2+t_0(n)}^{T+t_0(n)} \Big[\langle s^2_{i,t}\rangle - \big(\mathbf{f}^n_{i,\mathtt{new}}\big)^\top \langle s_{i,t}\mathbf{s}_{i,t-1}\rangle\Big], \end{aligned}$$

where $t_0(n) = (n-1)T$. Furthermore:

$$\begin{aligned} \mathbf{A}_{\mathtt{new}} &= \Big[\sum_{t=1}^{NT} \mathbf{x}_t \langle \mathbf{s}_t\rangle^\top\Big]\Big[\sum_{t=1}^{NT} \langle \mathbf{s}_t(\mathbf{s}_t)^\top\rangle\Big]^{-1}, \\ \mathbf{R}_{\mathtt{new}} &= \frac{1}{NT}\sum_{t=1}^{NT} \mathrm{diag}[\mathbf{x}_t(\mathbf{x}_t)^\top - \mathbf{A}_{\mathtt{new}}\langle \mathbf{s}_t\rangle(\mathbf{x}_t)^\top], \end{aligned}$$

where the diag$[\cdot]$ operator extracts the diagonal elements of the matrix. Following an M-step, the solution corresponding to $||\mathbf{A}_i|| = 1\ \forall i$ is chosen, where $||\cdot||$ is the Frobenius norm and $\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{i1} & \mathbf{a}_{i2} & \cdots & \mathbf{a}_{iQ} \end{bmatrix}^\top$, meaning that $\mathbf{A}$ and $\mathbf{Q}^n$ are scaled accordingly.

4.1.2 HARMONIC AND NOISE MODEL

The linear source parameters and signals are grouped as

$$\mathbf{d}^n_i \equiv \begin{bmatrix} (\mathbf{f}^n_i)^\top & (\mathbf{c}^n_i)^\top \end{bmatrix}^\top \quad , \quad \mathbf{z}_i \equiv \begin{bmatrix} (\mathbf{s}_{i,t-1})^\top & (\mathbf{u}_{i,t})^\top \end{bmatrix}^\top,$$

where

$$\mathbf{f}_i^n \equiv \begin{bmatrix} f_{i,1}^n & f_{i,2}^n & \cdots & f_{i,p}^n \end{bmatrix}^\top \quad , \quad \mathbf{c}_i^n \equiv \begin{bmatrix} c_{i,1} & c_{i,2}^n & \cdots & c_{i,p}^n \end{bmatrix}^\top .$$

It is in general not trivial to maximize $\mathcal{J}(\theta, \hat{p})$ with respect to $\omega_{i,0}^n$, since several local maxima exist, for example, at multiples of $\omega_{i,0}^n$ (McAulay and Quateri, 1986). However, simple grid search in a region provided satisfactory results. For each point in the grid we optimize $\mathcal{J}(\theta, \hat{p})$ with respect to $\mathbf{d}_i^n$:

$$\mathbf{d}_{i,\mathbf{new}}^n = \left[\sum_{t=2}^{NT} \left\langle \mathbf{z}_{i,t} (\mathbf{z}_{i,t})^\top \right\rangle \right]^{-1} \sum_{t=2}^{NT} \left\langle \mathbf{z}_{i,t} (s_{i,t})^\top \right\rangle .$$

The estimators of $\mathbf{A}$, $\mathbf{R}$ and q_i^n are similar to those in the AR model.

4.2 Gradient-based Learning

The derivative of the log-likelihood, $\frac{d\mathcal{L}(\theta)}{d\theta}$, can be computed and used in a quasi-Newton (QN) optimizer as is demonstrated in Olsson et al. (2006). The computation reuse the analysis of the M-step. This can be realized by rewriting $\mathcal{L}(\theta)$ as in Salakhutdinov et al. (2003):

$$\frac{d\mathcal{L}(\theta)}{d\theta} = \int p(\mathbf{S}|\mathbf{X}, \theta) \frac{d \log p(\mathbf{X}, \mathbf{S}|\theta)}{d\theta} d\mathbf{S} = \frac{d\mathcal{J}(\theta, \hat{p})}{d\theta}. \tag{5}$$

Due to the definition of $\mathcal{J}(\theta, \hat{p})$, the desired gradient in (5) can be computed following an E-step at relatively little effort. Furthermore, the analytic expressions are available from the derivation of the EM algorithm, see appendix A for details. A minor reformulation of the problem is necessary in order to maintain non-negative variances. Hence, the reformulations $\Omega^2 = \mathbf{R}$ and $(\phi_i^n)^2 = q_i^n$ are introduced. Updates are devised for Ω and ϕ_i^n. The derivatives are

$$\begin{aligned}
\frac{d\mathcal{L}(\theta)}{d\mathbf{A}} &= -\mathbf{R}^{-1}\mathbf{A} \sum_{t=1}^{NT} \left\langle \mathbf{s}_t \mathbf{s}_t^\top \right\rangle + \mathbf{R}^{-1} \sum_{t=1}^{N} \mathbf{x}_t \left\langle \mathbf{s}_t^\top \right\rangle , \\
\frac{d\mathcal{L}(\theta)}{d\Omega} &= \Omega^{-3} \sum_{t=1}^{NT} \left[\mathbf{x}_t \mathbf{x}_t^\top + \mathbf{A} \left\langle \mathbf{s}_t \mathbf{s}_t^\top \right\rangle \mathbf{A}^\top - 2\mathbf{x}_t \left\langle \mathbf{s}_t^\top \right\rangle \mathbf{A}^\top \right] , \\
\frac{d\mathcal{L}(\theta)}{d\mathbf{f}_i^n} &= \sum_{t=2+t_0(n)}^{T+t_0(n)} \left[\langle s_{i,t} \mathbf{s}_{i,t-1} \rangle - \left\langle \mathbf{s}_{i,t-1} \mathbf{s}_{i,t-1}^\top \right\rangle \mathbf{f}_i^n / q_i^n \right] , \\
\frac{d\mathcal{L}(\theta)}{d\phi_i^n} &= (1-T)/\phi_i^n + \\
&\quad \phi_i^{-3} \sum_{t=2+t_0(n)}^{T-1+t_0(n)} \left[\left\langle s_{i,t} s_{i,t}^\top \right\rangle + (\mathbf{f}_i^n)^\top \left\langle \mathbf{s}_{i,t-1} \mathbf{s}_{i,t-1}^\top \right\rangle \mathbf{f}_i^n - 2(\mathbf{f}_i^n)^\top \left\langle s_{i,t} \mathbf{s}_{i,t-1}^\top \right\rangle \right] .
\end{aligned}$$

In order to enforce the unit L2 norm on $\mathbf{A}_i$, a Lagrange multiplier is added to the derivative of $\mathbf{A}$. In this work, the QN optimizer of choice is the BFGS optimizer of Nielsen (2000).

4.3 Stochastic Gradient Learning

Although quasi-Newton algorithms often converge rapidly with a high accuracy, they do not scale well with the number of blocks, N. This is due to the fact that the number of parameters is asymptotically proportional to N, and therefore the internal inverse Hessian approximation becomes increasingly inaccurate. In order to be able to efficiently learn θ_2 ($\mathbf{A}$ and $\mathbf{R}$) for large N, a stochastic gradient approach (SGA), (Robbins and Monro, 1951), is employed.

It is adapted here to estimation in block-based state-space models, considering only a single randomly and uniformly sampled block, n, at any given time. The likelihood term corresponding to block n is $\mathcal{L}(\theta_1^n, \theta_2)$, where $\theta_1^n = \{\mathbf{F}^n, \mathbf{C}^n, \mathbf{Q}^n, \psi^n\}$. The stochastic gradient update to be applied is computed at the current optimum with respect to θ_1^n,

$$\begin{aligned} \Delta\theta_2 &= \eta \frac{d\mathcal{L}(\hat{\theta}_1^n, \theta_2)}{d\theta_2}, \\ \hat{\theta}_1^n &= \arg\max_{\theta_1^n} \mathcal{L}(\theta_1^n, \theta_2). \end{aligned}$$

where $\hat{\theta}_1^n$ is estimated using the EM algorithm. Employing an appropriate 'cooling' of the learning rate, η, is mandatory in order to ensure convergence: one such, devised by Robbins and Monro (1951), is choosing η proportional to $\frac{1}{k}$ where k is the iteration number. In our simulations, the SGA seemed more robust to the initial parameter values than the QN and the EM algorithms.

5. Learning from Synthetic Data

In order to investigate the convergence of the algorithms, AR(2) processes with time-varying pole placement were generated and mixed through randomly generated filters. For each signal frame, $T = 200$, the poles of the AR processes were constructed so that the amplification, r, was fixed while the center frequency was drawn uniformly from $\mathcal{U}(\pi/10, 9\pi/10)$. The filter length was $L = 8$ and the coefficients of the mixing filters, that is, the $\mathbf{a}_{ij}$ of $\mathbf{A}$, were generated from i.i.d. Gaussians weighted by an exponentially decaying function. Quadratic mixtures with $Q = P = 2$ were used: the first 2 elements of $\mathbf{a}_{12}$ and $\mathbf{a}_{21}$ were set to zero to simulate a situation with different channel delays. All channel filters were normalized to $||\mathbf{a}_{ij}||_2 = 1$. Gaussian i.i.d. noise was added in each channel, constructing the desired signal to noise ratio.

For evaluation purposes, the signal-to-error ratio (SER) was computed for the inferred sources. The true and estimated sources were mapped to the output by filtering through the direct channel so that the true source at the output is $\tilde{s}_{i,t} = \mathbf{a}_{ii} * s_{i,t}$. Similarly defined, the estimated source at the sensor is $\hat{s}_{i,t}$. Permutation ambiguities were resolved prior to evaluating the SER,

$$\mathrm{SER}_i = \frac{\sum_t \tilde{s}_{i,t}^2}{\sum_t (\tilde{s}_{i,t} - \hat{s}_{i,t})^2}.$$

The EM and QN optimizers were applied to learn the parameters from $N = 10$ frames of samples with SNR $= 20$dB, $r = 0.8$. The algorithms were restarted 5 times with random initializations, $\mathbf{A}_{ij} \in \mathcal{N}(0,1)$, the one that yielded the maximal likelihood was selected. Figure 2 shows the results of the EM run: the close match between the true and learned models confirms that the parameters can indeed be learned from the data using maximum-likelihood optimization. In Table 1, the generative approach is contrasted with a stationary finite impulse response (FIR) filter separator that

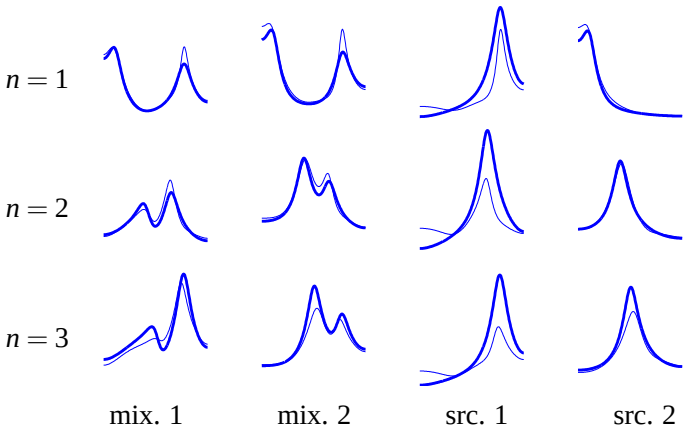

Figure 2: The true (bold) and estimated models for the first 3 frames of the synthetic data based on the autoregressive model. The amplitude frequency responses of the combined source and channel filters are shown: for source i, this amounts to the frequency response of the filter, with the scaling and poles of $\theta_{1,i}$ and zeros of the direct channel $\mathbf{a}_{ii}$. For the mixtures, the responses across channels were summed. The EM algorithm provided the estimates.

	Estimated	Generative	MSE FIR
AR	9.1 ± 0.4	9.7 ± 0.4	7.5 ± 0.2
HN	11.8 ± 0.7	13.2 ± 0.4	7.9 ± 0.5

Table 1: The signal-to-error ratio (SER) performance on synthetic data based on the autoregressive (AR) and harmonic-and-noise (HN) source models. Mean and standard deviation of the mean are shown for 1) the EM algorithm applied to the mixtures, 2) inferences from data and the true model, and, 3) the optimal FIR filter separator. The mean SER and the standard deviation of the mean were calculated from $N = 10$ signal frames, SNR $= 20$dB.

in a supervised fashion was optimized to minimize the squared error between the estimated and true sources, $L_{\text{FIR}} = 25$. Depending on the signal properties, the generative approach, which results in a time-varying filter, results in a clear advantage over the time-invariant FIR filter, which has to compromise across the signal's changing dynamics. As a result, the desired signals are only sub-optimally inferred by methods that apply a constant filter to the mixtures. The performance of the learned model is upper-bounded by that of the generative model, since the maximum likelihood estimator is only unbiased in the limit.

The convergence speed of the EM scheme is highly sensitive to the signal-to-noise ratio of the data, as was documented in Olsson et al. (2006), whereas the QN algorithm is more robust to this condition. In Bermond and Cardoso (1999), it was shown that the magnitude of the update of $\mathbf{A}$ scales inversely with the SNR. By varying the SNR in the synthetic data and applying the EM algorithm, it was confirmed that the predicted convergence slowdown occurs at high SNR. In contrast, the QN algorithm was found to be much more robust to the noise conditions of the data. Figure 3 shows the SER performance of the two algorithms as computed following a fixed number

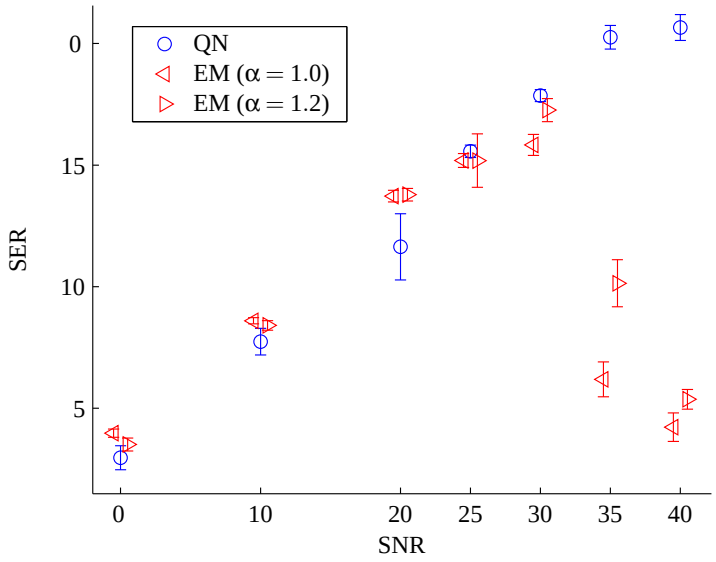

Figure 3: Convergence properties of the EM and QN algorithms as measured on the synthetic data (autoregressive sources). The signal-to-error ratio (SER) was computed in a range of SNR following 300 iterations. As the SNR increases, more accurate estimates are provided by all algorithms, but the number of iterations required increases more dramatically for the EM algorithm. Results are shown for the basic EM algorithm as well as for the step-size adjusted version.

of iterations ($i_{\max} = 300$). It should be noted that the time consumption per iteration is similar for the two algorithms, since a similar number of E-step computations is used (and E-steps all but dominate the cost).

For the purpose of analyzing the HN model, a synthetic data set was generated. The fundamental frequency of the harmonic component was sampled uniformly in a range, see Figure 4, amplitudes and phases, $K = 4$, were drawn from a Gaussian distribution and subsequently normalized such that $||\mathbf{c}_i|| = 1$. The parameters of the model were estimated using the EM algorithm on data, which was constructed as SNR $= 20$dB, HNR $= 20$dB. The fundamental frequency search grid was defined by 101 evenly spaced points in the generative range. In Figure 4, true and learned parameters are displayed. A close match between the true and estimated harmonics is observed.

In cases when the sources are truly harmonic and noisy, it is expected that the AR model performs worse than the HN model. This is due to the fact that a harmonic mean structure is required for the model to be unbiased. The AR model will compensate by estimating a larger variance, q_i, leading to suboptimal inference. In Figure 5, the bias is quantified by measuring the performance gap between the HN and AR models for varying HNR. The source parameters were estimated by the EM algorithm, whereas the mixing matrix, $\mathbf{A}$, was assumed known.

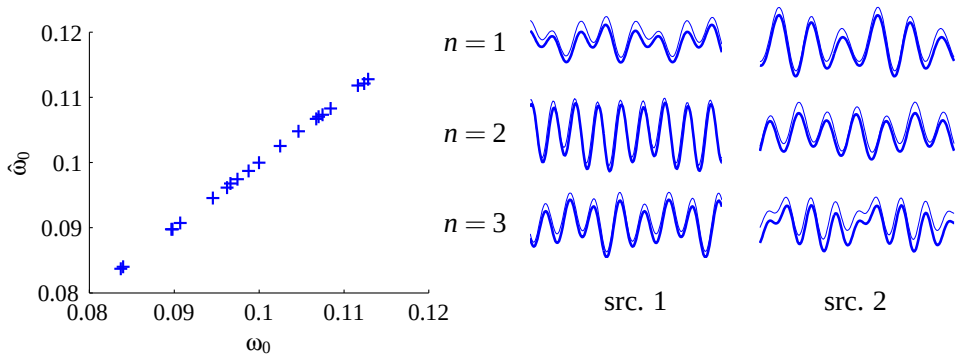

Figure 4: The true and estimated parameters from synthetic mixtures of harmonic-and-noisy sources as obtained by the EM algorithm. Left: fundamental frequencies in all frames. Right: the waveforms of the true (bold) and estimated harmonic components. For visualization purposes, the estimated waveform was shifted by a small offset.

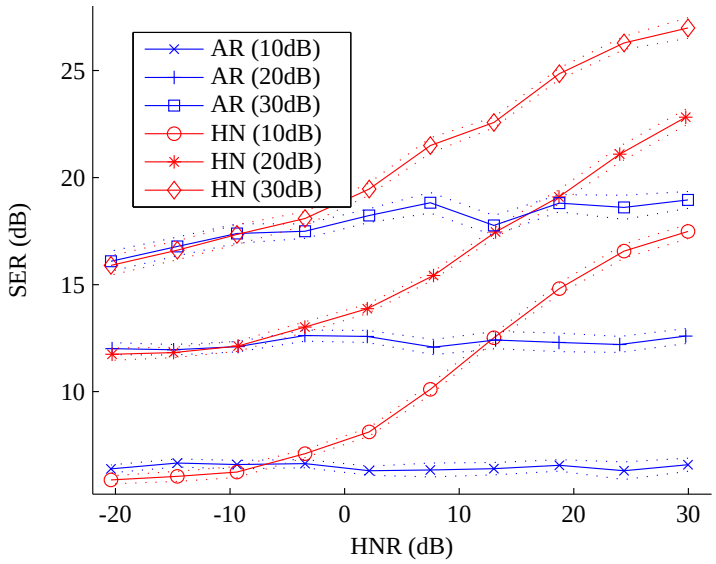

Figure 5: The signal-to-error ratio (SER) performance of the autoregressive (AR) and harmonic-and-noisy (HN) models for the synthetic data set ($N = 100$) in which the harmonic-to-noise ratio (HNR) was varied. Results are reported for SNR $= 10, 20, 30$dB. The results indicate that the relative advantage of using the correct model (HN) can be significant. The error-bars represent the standard deviation of the mean.

6. Speech Mixtures

The separation of multiple speech sources from room mixtures has potential applications in hearing aids and speech recognition software (see, for example, Parra and Spence, 2000). For this purpose,

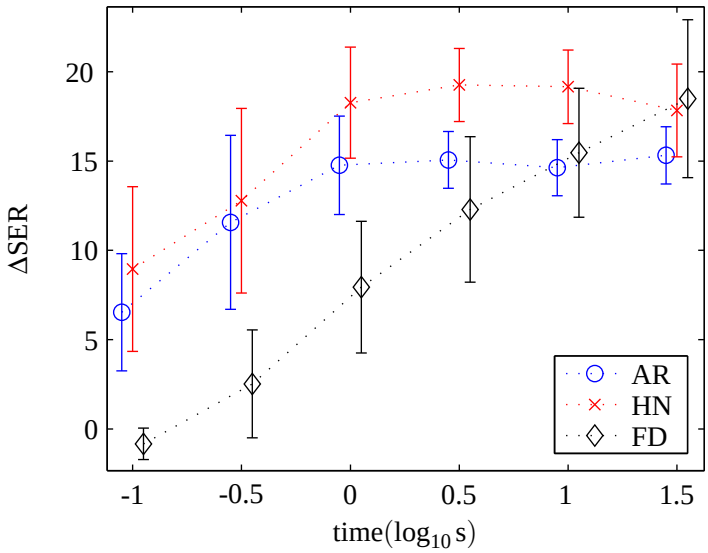

Figure 6: The separation performance (SER) on test mixtures as a function of the training data duration for the autoregressive (AR) and harmonic-and-noisy (HN) priors. Using the stochastic gradient (SG) algorithm, the parameters were estimated from the training data. Subsequently, the learned filters, $\mathbf{A}$, were applied to the test data, reestimating the source model parameters. The noise was constructed at 40dB and assumed known. For reference, a frequency domain (FD) blind source separation algorithm was applied to the data.

we investigate the models based on the autoregressive (AR) and harmonic-and-noisy source (HN) priors and compare with a standard frequency domain method (FD). More specifically, a learning curve was computed in order to illustrate that the inclusion of prior knowledge of speech benefits the separation of the speech sources. In Figure 6 is shown the relationship between the separation performance on test mixtures and the duration of the training data, confirming the hypothesis that the AR and HN models converge faster than the FD method. Furthermore it is seen that the HN model can obtain a larger SER than the AR model.

The mixtures were constructed by filtering speech signals (sampled at 8Hz) through a set of simulated room impulse responses, that is, $\mathbf{a}_{ij}$, and subsequently adding the filtered signals. The room impulse responses were constructed by simulating $Q = 2$ speakers and $P = 2$ microphones in an (ideal) anechoic room, the cartesian coordinates in the horizontal plane given (in m) by $\{(1,3),(3,3)\}$ and $\{(1.75,1),(2.25,1)\}$ for the speakers and microphones, respectively.[1]. This corresponds to a maximum distance of 1.25m between the speakers and the microphones, and a set of room impulse responses that are essentially Kronecker delta functions well represented using a filter length of $L = 8$.

1. A Matlab function, rir.m, implementing the image method (Allen and Berkley, 1979) is available at http://2pi.us/rir.html.

The SG algorithm was used to fit the model to the mixtures and subsequently infer the source signals. The speech data, divided into blocks of length $T = 200$, was preprocessed with a standard pre-emphasis filter, $H(z) = 1 - 0.95z^{-1}$, and inversely filtered prior to the SER calculations. From initial conditions ($q_i^n = 1$, $f_{i,j}^n = 0$, $c_{i,j}^n = 0$ and $a_{i,j,k}$ normally distributed, variance 0.01, for all i, j, n, k except $a_{1,1,1} = 1$, $a_{2,2,1} = 1$; $\omega_{0,i}^n$ was drawn from a uniform distribution corresponding to the interval $50 - 200$Hz), the algorithm was allowed $i_{\max} = 500$ iterations to converge and restarts were not necessary. The source model order was set to $p = 1$ (autoregression order) and in the case of the harmonic-and-noise model, the number of harmonics was set to $K = 6$. The complex JADE algorithm was employed in the frequency domain as the reference method (Cardoso and Souloumiac, 1993). In order to correct the permutations across the 101 frequencies, amplitude correlation between the bands was maximized (see, for example, Olsson and Hansen, 2006).

In order to qualitatively assess the effect of the two priors, a mixture of speech signals was constructed using $P = 2$ speech signals (a female and a male, shown in Figure 7a and b). They were mixed through artificial channels, $\mathbf{A}$, which were generated as in Section 5. Noise was added up to a level of 20dB. The EM algorithm was used to fit the source models to the mixtures. It is clear from Figure 7 c-f that the estimated harmonic model to a large extent explains the voiced parts of the speech signals, and the unvoiced parts to a lesser extent. In regions of rapid fundamental frequency variation, the harmonic part cannot be fitted as well (the frames are too long here). In Figure 7 g and h, the separation performances of the AR and HN models are contrasted. Most often, the HN performs better than the AR model. A notable exception occurs in the case when either speaker is silent, in which case the misfit of the HN model is more severe, suggesting that the performance can be improved by model control.

7. Conclusion

It is demonstrated that careful generative modelling is a viable approach to convolutive source separation and can yield improved results. Noisy observations, non-stationarity of the sources and small data volumes are examples of scenarios which benefit from the higher level of modelling detail.

The performance of the model was shown to depend on the choice of optimization scheme when the signal-to-noise ratio is high. In this case, the EM algorithm, which is often preferable for its conceptual and analytical simplicity, experiences a substantial slowdown, and alternatives must be employed. Such an alternative is a gradient-based quasi-Newton algorithm, which is shown to be particularly useful in low-noise settings. Furthermore, the required gradients are obtained in the process of deriving the EM algorithm.

The harmonic-and-noise model was investigated as a means to estimating more accurately a number of speech source signals from the their mixtures. Although a substantial improvement is shown to result when the sources are truly harmonic, the overall model is vulnerable to overfitting when the energy of one or more sources is locally near-zero. An improvement of the existing framework would be a model control scheme, such as variational Bayes, which could potentially cancel the negative impact of speaker silence. This is a topic for future research.

Acknowledgments

The authors thank the Oticon Fonden for providing financial support.

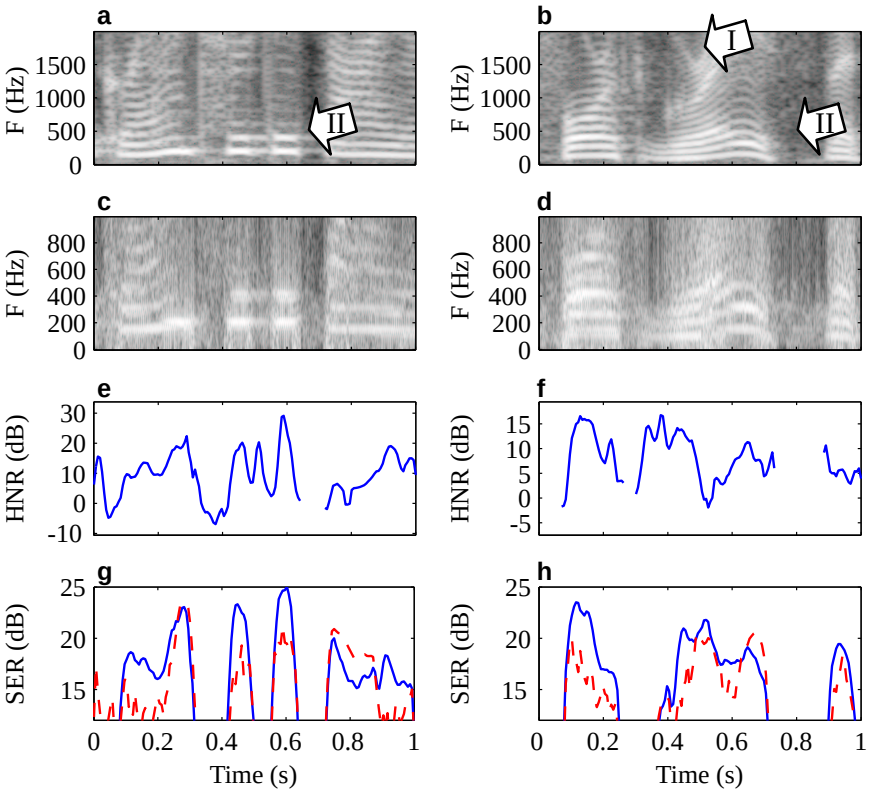

Figure 7: The source parameters of the autoregressive (AR) and harmonic-and-noisy (HN) models were estimated from $Q = 2$ convolutive mixtures using the EM algorithm. Spectrograms show the low-frequent parts of the original female (a) and male (b) speech sources. The appropriateness of the HN model can be assessed in c and d, which displays the re-synthesization of the two sources from the parameters ($K = 6$), as well as e and f, where the estimated ratio of harmonics to noise (HNR) is displayed. Overall the fit seem good, except at rapid variations of the fundamental frequency, for example, at (I), where the analysis frames are too long. The relative separation performance of the AR and HN models, which is shown in g and h for the two sources, confirms that the HN model is superior in most cases, with a notable exception in regions such as (II), where one of the speakers is silent. This implies a model complexity mismatch which is more severe for the more complex HN model.

Appendix A.

Below, an example of an M-step update derivation is shown for $\mathbf{F}^n$. As a by-product of the analysis, the derivative for the gradient-based optimizers appears. Care must be devised in obtaining the derivatives, since $\mathbf{F}^n$ is a structured matrix, for example, certain elements are one and zero. Therefore, the cost-function is expressed in terms of $\mathbf{f}_i^n$ rather than $\mathbf{F}^n$. Since all variables, which are here

indexed by the block identifier, n, are Gaussian, we have that:

$$\begin{aligned}\mathcal{J}(\theta) &= -\frac{1}{2}\sum_{n=1}^{N}\Big[\sum_{i=1}^{P}\Big\{\log|\Sigma_i^n| + \Big\langle\left(\mathbf{s}_{i,1}^n-\mu_i^n\right)^T\left(\Sigma_i^n\right)^{-1}\left(\mathbf{s}_{i,1}^n-\mu_i^n\right)\Big\rangle\Big\} \\ &\quad +(T-1)\sum_{i=1}^{P}\log q_i^n + \frac{1}{q_i^n}\sum_{t=2}^{T}\sum_{i=1}^{P}\Big\langle\left(s_{i,t}^n-\left(\mathbf{f}_i^n\right)^T\mathbf{s}_{i,t-1}^n\right)^2\Big\rangle \\ &\quad +T\log\det\mathbf{R}+\sum_{t=1}^{T}\Big\langle\left(\mathbf{x}_t^n-\mathbf{A}\mathbf{s}_t^n\right)^T\mathbf{R}^{-1}\left(\mathbf{x}_t^n-\mathbf{A}\mathbf{s}_t^n\right)\Big\rangle\Big].\end{aligned}$$

The vector derivative of $\mathcal{J}(\theta)$ with respect to $\mathbf{f}_i^n$ is:

$$\frac{d\mathcal{J}(\theta)}{d\mathbf{f}_i^n} = \frac{1}{q_i^n}\left[\sum_{t=2}^{T}\Big\langle\mathbf{s}_{i,t-1}^n\left(\mathbf{s}_{i,t-1}^n\right)^\top\Big\rangle\mathbf{f}_i^n-\sum_{t=2}^{T}\left\langle\mathbf{s}_{i,t-1}^n s_{i,t}^n\right\rangle\right].$$

This was the desired gradient, which is directly applicable in a gradient-based algorithm. By equating to zero and solving, the M-step update is derived:

$$\mathbf{f}_{i,\mathtt{new}}^n = \left[\sum_{t=2}^{T}\Big\langle\mathbf{s}_{i,t-1}^n\left(\mathbf{s}_{i,t-1}^n\right)^\top\Big\rangle\right]^{-1}\sum_{t=2}^{T}\left\langle\mathbf{s}_{i,t-1}^n s_{i,t}^n\right\rangle.$$

References

J. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65:943–950, 1979.

J. Anemüller and B. Kollmeier. Amplitude modulation decorrelation for convolutive blind source separation. In *Proc. ICA 2000*, pages 215–220, 2000.

A J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

O. Bermond and J.-F. Cardoso. Approximate likelihood for noisy mixtures. In *Proc. ICA*, pages 325–330, 1999.

M. Brandstein. On the use of explicit speech modeling in microphone array applications. In *Proc. ICASSP*, pages 3613–3616, 1998.

J.-F. Cardoso, H. Snoussi, J. Delabrouille, and G. Patanchon. Blind separation of noisy Gaussian stationary sources. Application to cosmic microwave background imaging. In *Proc. EUSIPCO*, pages 561–564, 2002.

J. F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings F*, 140(6):362–370, 1993.

P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistics Society, Series B*, 39:1–38, 1977.

M. Dyrholm and L. K. Hansen. CICAAR: Convolutive ICA with an auto-regressive inverse model. In *Proc. ICA 2004*, pages 594–601, 2004.

P. A. Højen-Sørensen, Ole Winther, and Lars Kai Hansen. Mean-field approaches to independent component analysis. *Neural Computation*, 14(4):889–918, 2002.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, Inc, 2001.

R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering ASME Transactions*, 83:95–107, 1960.

T.W. Lee, A. J. Bell, and R. H. Lambert. Blind separation of delayed and convolved sources. In *Advances of Neural Information Processing Systems*, volume 9, page 758, 1997.

R.J. McAulay and T.F. Quateri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(4):744–754, 1986.

M. McKeown, L.K. Hansen, and T.J. Sejnowski. Independent component analysis for fmri: What is signal and what is noise? *Current Opinion in Neurobiology*, 13(5):620–629, 2003.

E. Moulines, J. Cardoso, and E. Cassiat. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proc. ICASSP*, volume 5, pages 3617–3620, 1997.

H. B. Nielsen. UCMINF - an algorithm for unconstrained nonlinear optimization. Technical Report IMM-Rep-2000-19, Technical University of Denmark, 2000.

R. K. Olsson and L. K. Hansen. A harmonic excitation state-space approach to blind separation of speech. In *Advances in Neural Information Processing Systems*, volume 17, pages 993–1000, 2005.

R. K. Olsson and L. K. Hansen. Blind separation of more sources than sensors in convolutive mixtures. In *International Conference on Acoustics, Speech and Signal Processing*, 2006.

R. K. Olsson, K. B. Petersen, and T. Lehn-Schiøler. State-space models - from the EM algorithm to a gradient approach. *Neural Computation - to appear*, 2006.

L. Parra and C. Spence. Convolutive blind separation of non-stationary sources. *IEEE Transactions, Speech and Audio Processing*, pages 320–7, 5 2000.

B. A. Pearlmutter and L. C. Parra. A context-sensitive generalization of ICA. In *In Advances in Neural Information Processing Systems 9*, pages 613–619, 1997.

H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

R. Salakhutdinov and S. Roweis. Adaptive overrelaxed bound optimization methods. In *International Conference on Machine Learning*, pages 664–671, 2003.

R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani. Optimization with EM and Expectation-Conjugate-Gradient. In *International Conference on Machine Learning*, volume 20, pages 672–679, 2003.

R. Shumway and D. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Anal.*, 3(4):253–264, 1982.

E. Weinstein, M. Feder, and A. V. Oppenheim. Multi-channel signal separation by decorrelation. *IEEE Transactions on Speech and Audio Processing*, 1(4), 1993.

D. Yellin and E. Weinstein. Multichannel signal separation: Methods and analysis. *IEEE Transactions on Signal Processing*, 44(1):106–118, 1996.

Appendix IX

R. K. Olsson and L. K. Hansen, Blind Separation of More Sources than Sensors in Convolutive Mixtures, International Conference on Acoustics on Speech and Signal Processing, 5, 657-660, 2006

BLIND SEPARATION OF MORE SOURCES THAN SENSORS IN CONVOLUTIVE MIXTURES

Rasmus Kongsgaard Olsson and Lars Kai Hansen

Informatics and Mathematical Modelling, Technical University of Denmark
2800 Lyngby, Denmark, rko,lkh@imm.dtu.dk

ABSTRACT

We demonstrate that blind separation of more sources than sensors can be performed based solely on the second order statistics of the observed mixtures. This a generalization of well-known robust algorithms that are suited for equal number of sources and sensors. It is assumed that the sources are non-stationary and sparsely distributed in the time-frequency plane. The mixture model is convolutive, i.e. acoustic setups such as the cocktail party problem are contained. The limits of identifiability are determined in the framework of the PARAFAC model. In the experimental section, it is demonstrated that real room recordings of 3 speakers by 2 microphones can be separated using the method.

1. INTRODUCTION

The human auditory system solves the so-called cocktail party problem, i.e it separates out a single speech signal from a composition of speech signals and possibly other interfering noises. This is an instance of blind source separation (BSS). Machines capable of emulating this function have potential applications in e.g. hearing aids and audio communication. The *convolutive* mixture model accounts for the various delays and attenuations of e.g. an acoustic mixture:

$$\mathbf{x}[t] = \sum_{k=0}^{L-1} \mathbf{A}[k]\mathbf{s}[t-k] \tag{1}$$

where $\mathbf{x}[t]$ and $\mathbf{s}[t]$ are an N dimensional sensor vector and an R dimensional source vector, respectively, sampled at discrete time t. The matrices $\mathbf{A}[k]$ contain the impulse responses of the signal channels. The sources can only be recovered blindly, i.e. $\mathbf{A}[k]$ unknown, up to an arbitrary scale and permutation of source index.

In many cases, such as the cocktail party situation where the speakers are independent on the timescale of interest, the problem can to some extend be solved by algorithms that are based on independent component analysis (ICA), [1]. In particular, the instantaneous mixture model, which arises as a results of $L = 1$, is a well-solved problem, see, e.g., [2]. However, this mixing model ($L = 1$) is inappropriate and insufficient for the separation of acoustically mixed audio signals for the reasons already mentioned. ICA algorithms determine $\mathbf{s}_t$ by assuming statistical independency of the sources and certain properties of the distribution of $\mathbf{s}_t$, where non-Gaussianity, non-stationarity and non-whiteness are the most important. Convolutive ICA algorithms, i.e. $L > 1$, have been devised by e.g. [3] and [4]. The most efficient methods use transformation to the discrete Fourier domain, where convolution approximately translates to multiplication, yielding a separate instantaneous ICA problem for each evaluated frequency. As a result, the resulting arbitrary permutations across frequency problem must be resolved. Algorithms that function in the time-domain, such as [5], can benefit more directly from domain-oriented source modelling, but typically at a higher computational cost.

Common to the algorithms mentioned above are that they assume *quadratic* mixtures, that is, the number of sensors equals the number of sources, or $N = R$. The class of mixtures, where $R > N$, are termed *underdetermined*.[1] Instantaneous ICA algorithms have been devised to solve the underdetermined problem, i.e. [6], [7] and [8]. These methods assume a sparse distribution of the sources, either directly in the time-domain or in a transformed domain.

An alternative approach to blind source separation is the use of binary masks in the spectrogram, i.e. assigning each point in the time-frequency plane to a source. The masks are often constructed using segmentation cues inspired by the human auditory system, such as interaural intensity and time differences (IID/ITD), [9]. Efforts to combine ICA with binary masks have been undertaken by [10]. A problem introduced by binary masks is that artifacts, or unnatural sounds, may appear in the reconstructed signals.

The major contribution of this work is to generalize to the overcomplete case the robust algorithms of [3] and [13], which handle quadratic mixtures relying solely on robust time-varying second order statistics in the power spectral domain. Since the essentially non-stationary Gaussian signal model is an instance of the trilinear PARAFAC[2] model, [11], the results from this field are employed to construct the algorithm and certify the identifiability of the model under various assumptions. One observation derived from the PARAFAC formulation is that the source power spectra are identifiable for ($N = 2$, $R = 3$) provided that the mixing process parameters ($\mathbf{A}[k]$) are available at that stage. As a consequence, the maximum posteriori estimates of the sources can be computed as opposed to the usual binary mask reconstructions. A key component in determining $\mathbf{A}[k]$ is the sparsity of the sources in the time-frequency plane, which allows for estimation of this part of the model through k-means clustering, [8]. As evidence of the usefulness of the approach, it is demonstrated that speech mixtures ($N = 2$, $R = 3$) recorded in a *real* office environment can be handled, see `www.imm.dtu.dk/~rko/underdetermined`.

In section 2 and 3, the PARAFAC formulation of the blind source separation problem is motivated. In sections 4 and 6, the estimation of the parameter and source inference, respectively, are covered. The limits of identifiability are discussed in section 5. Implementation issues are summarized in section 7. The performance of the algorithm

Thanks to Oticon fonden for financially supporting this work.

[1] One-sensor separation is a topic in its own right and is not discussed here.

[2] Parallel Factor Analysis. Also known as Canonical Decomposition (CANDECOMP).

is gauged on benchmark audio data in the experimental section.

2. PARALLEL FACTOR ANALYSIS

A thorough review of the following theory can be found in [12]. Consider the 3-way array x_{ijk} indexed by $i \in [1,..,I]$, $j \in [1,..,J]$, $k \in [1,..,K]$. The trilinear PARAFAC decomposition is defined:

$$x_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} c_{kf}$$

with loading matrices $(\mathbf{A})_{if} = a_{if}$, $(\mathbf{B})_{jf} = b_{jf}$ and $(\mathbf{C})_{kf} = c_{kf}$. The PARAFAC model can equivalently be expressed in terms its matrices or 'slabs':

$$\mathbf{X}_k = \mathbf{A}\text{diag}_k[\mathbf{C}]\mathbf{B}^\top \tag{2}$$

where $\text{diag}_k[\cdot]$ operating on a matrix constructs a diagonal matrix with the kth row of the matrix as diagonal elements. The PARAFAC model could equivalently be expressed along $\mathbf{A}$ or $\mathbf{B}$. The *matrization* of the PARAFAC model is yet another representation: $\mathbf{X}_{IJ\times K} = (\mathbf{B} \odot \mathbf{A})\,\mathbf{C}^\top$, where the Khatri-Rao product is defined

$$(\mathbf{B} \odot \mathbf{A}) \equiv \begin{bmatrix} \mathbf{A}\text{diag}_1[\mathbf{B}] \\ \vdots \\ \mathbf{A}\text{diag}_J[\mathbf{B}] \end{bmatrix}$$

The indices of $\mathbf{X}_{IJ\times K}$ indicate the direction and hence the dimensions of the matrization. The former index varies more rapidly. A sufficient condition for uniqueness was provided by Kruskal in [11]:

$$k[\mathbf{A}] + k[\mathbf{B}] + k[\mathbf{C}] \geq 2(F-1) \tag{3}$$

where the k-rank, denoted $k[\mathbf{A}]$, of a matrix $\mathbf{A}$, is defined as the maximal integer m such that any m columns of $\mathbf{A}$ form a linearly independent set. Clearly, $k[\mathbf{A}] \leq r[\mathbf{A}]$, where $r[\mathbf{A}]$ is the rank of $\mathbf{A}$.

3. MODEL

The convolutive mixture of equation (1) will form the basis of the model. Only the autocorrelation functions of $\mathbf{s}_t$ are considered in the following analysis, which is similar to imposing a multivariate Gaussian model on $\{\mathbf{s}[t], \mathbf{s}[t+1], \ldots, \mathbf{s}[t+T_c-1]\}$, where T_c is the correlation length. The assumptions can be summarized as:
`A1`: the sources, $\mathbf{s}[t]$, are zero mean with no cross-correlation, i.e. $\langle \mathbf{s}[t]\mathbf{s}^\top[t-\tau] \rangle$ is a diagonal matrix for all τ.
`A2`: the signal channels, $\mathbf{A}[k]$, are constant on the time-scale of analysis. No columns are collinear, as this would effectively constitute a reduction of N.
`A3`: the autocorrelation functions $\langle \mathbf{s}[t]\mathbf{s}^\top[t-\tau] \rangle$ are time-varying as in [3]. The variation patterns are independent for each source.
`A4`: At most one source is non-zero in any time-frequency block $(\{n, \ldots, n+M-1\}, k)$, where M is the block length not to be confused with the frame length, K . This effectively is an assumption of sparsity as in, e.g., [8]. In figure 1, it is demonstrated that for certain quasi-periodic signals such as speech, the number of (n,k) bins required to represent a speech signal is indeed small.

The discrete Fourier transform (DFT) is applied to windowed frames of $\mathbf{x}_t$, length K, obtaining:

$$\mathbf{x}_k^{(n)} = \mathbf{A}_k \mathbf{s}_k^{(n)} + \mathbf{e}_k^{(n)} \tag{4}$$

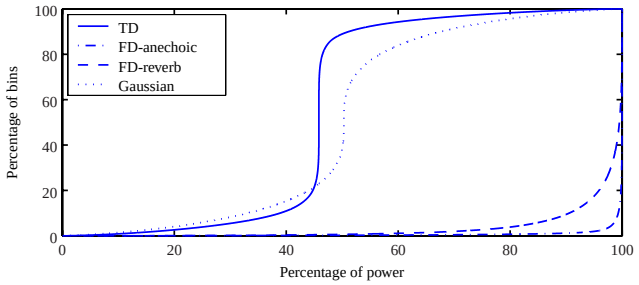

Fig. 1. The sparsity of speech ($F_s = 8kHz$) under various conditions presented as the minimal percentage of frequency bins required to represent a given percentage of power. The raw time-domain speech signal (TD) is shown as well as its time-frequency transformation (FD-anechoic) and a version with simulated large-room reverberation (FD-reverb). DFT lenght: $K = 512$, no overlap.

where $\mathbf{x}_k^{(n)}$, $\mathbf{A}_k$ and $\mathbf{s}_k^{(n)}$ are the DFT of the corresponding time-domain signals at discrete frequencies $k \in [0, 1, \ldots, K-1]$ and frame n. The residual term, $\mathbf{e}_k^{(n)}$, is due to equation (1) being a linear convolution rather than a circular one. When $L \ll K$, the mismatch vanishes, that is $\langle \frac{|\mathbf{e}_k|}{|\mathbf{x}_k|} \rangle \to 0$. Furthermore, the auto/cross power spectra of $\mathbf{x}_t$ as a function of frame index n, $\mathbf{C}_k^{(n)}$, can be computed from the power time-spectra of $\mathbf{s}_t$:

$$\mathbf{C}_k^{(n)} = \mathbf{A}_k \mathbf{D}_k^{(n)} \mathbf{A}_k^H + \mathbf{E}_k^{(n)} \tag{5}$$

where $\mathbf{D}_k^{(n)}$ is a diagonal matrix (due to `A1`) with the power of the sources in time-frequency bin (n,k) as diagonal elements. The channels, $\mathbf{A}_k$, are independent of n due to `A2`. The power spectrum residual, $\mathbf{E}_k^{(n)}$ can be neglected when $\mathbf{e}_k^{(n)}$ is small. As was also noted in [13], any linear channel that exhibits a sufficiently rapidly decaying autocorrelation function can be treated by our approach, not just a convolutive channel.

By comparing with equation (2), it is seen that equation (5) is approximately a PARAFAC model. The following reformulation of (5) is convenient:

$$\mathbf{C}_{NN\times K}[k] = (\mathbf{A}_k \odot \mathbf{A}_k^*)\,\mathbf{\Lambda}_k^\top \tag{6}$$

where $\mathbf{C}_{NN\times K}[k]$ is the matricized auto/cross power at frequency k and $(\mathbf{\Lambda}_k)_{nj}$ is the power of source j at time-frequency bin (n,k).

4. PARAMETER ESTIMATION

A standard approach to estimating the parameters of equation (6) is the alternating least squares algorithm (ALS), which alternatingly minimizes the Frobenius norm of $\mathbf{R}_{NN\times K}[k] - \mathbf{C}_{NN\times K}[k]$ with respect to the matrices $\mathbf{A}_k$ and $\mathbf{\Lambda}[k,n]$, where $\mathbf{R}_{NN\times K}[k]$ is the estimated auto/cross power spectra of $\mathbf{x}_t$, e.g.:

$$\hat{\mathbf{\Lambda}}_k^\top = \arg\min_{\mathbf{\Lambda}} \left\| \mathbf{R}_{NN\times K}[k] - (\mathbf{A}_k \odot \mathbf{A}_k^*)\mathbf{\Lambda}_k^\top \right\|^2 \tag{7}$$

The arbitrary scaling of the model is fixed by normalizing the columns of $\mathbf{A}_k$ so that $\|\mathbf{a}_{i,k}\|^2 = 1$ and zeroing the phases of the 1st row: $\angle(\mathbf{A})_{1i,k} = 0\ \forall i$. The solution to equation (7) is just a least-squares fit:

$$\hat{\mathbf{\Lambda}}_k^\top = (\mathbf{A}_k \odot \mathbf{A}_k^*)^\dagger\, \mathbf{R}_{NN\times K}[k] \tag{8}$$

	assumptions	bound
I	A1, A2, A3	$2N - 2 \geq R$
II	A1, A2, A3, A4	$\frac{1}{2}N(N+1) \geq R$

Table 1. Bounds of identifiability of the source spectrograms depending on the set of model assumptions.

where † is the pseudoinverse operator. An alternative means of estimating $\mathbf{A}_k$ and $\mathbf{D}[k,n]$ from $\mathbf{C}[k,n]$ and equation (5) is the application of a joint diagonalization algorithm such as in [14]. However, the ALS is conceptually simple and has good convergence properties under certain circumstances as was demonstrated in [13]. In the case of a 1-component model, i.e. when $\mathbf{A}_k$ and $\mathbf{\Lambda}_k$ consist of a single column ($\mathbf{a}_k$ and λ_k), a particularly simple solution exists:

$$\left[\sum_n \mathbf{R}_k^{(n)}\right] \mathbf{a}_k = \left[\sum_n \left(\lambda_k^{(n)}\right)^2\right] \mathbf{a}_k \tag{9}$$

where $\mathbf{R}_k^{(n)}$ and $\lambda_k^{(n)}$ are the measured auto/cross power of $\mathbf{x}_t$ and the power of $\mathbf{s}_t$ in time-frequency bin (n,k), respectively. The $\mathbf{a}_k$ corresponding to the maximal eigenvalue is the least-squares estimate.

5. IDENTIFIABILITY

In the following will be discussed mainly the limits of recovering the time-varying source power spectra, $\mathbf{\Lambda}_k$, i.e. the spectrograms. The MAP inference of $\mathbf{s}_k^{(n)}$ is treated in section 6. It was mentioned in section 1 that $\mathbf{\Lambda}_k$ can only be blindly recovered up to an unknown scaling and ordering, i.e. it is only possible to estimate $\mathbf{P}_k\mathbf{H}_k\mathbf{\Lambda}_k$, where $\mathbf{H}_k$ and $\mathbf{P}_k$ are (diagonal) scaling and permutation matrices, respectively. The frequency permutation problem, i.e. estimating $\mathbf{P}_k$ for all k, can be remedied by defining a similarity measure across frequencies as in e.g. [13]. The scaling matrix was fixed in the parameter estimation process due to certain assumptions about the scale of $\mathbf{A}_k$.

The identifiability of the source spectrograms, $\mathbf{\Lambda}_k$, is determined in the general blind case ($\mathbf{A}_k$ is unknown) under assumptions A1, A2 and A3. The uniqueness theorem (3) yields a lower bound for identifiability:

$$2N \geq R + 2$$

where the full rank of $\mathbf{A}_k$ and $\mathbf{\Lambda}_k$ was assumed, consequences of A2 and A3, respectively. This means that $\mathbf{\Lambda}_k$ can be estimated in many cases where $R > N$, however notably excluding the $N = 2$, $R = 3$ case. In [15], it was shown that the bound is tight for $R = 3$, but not necessarily for $R > 3$.

In order to retrieve the source power spectrograms in the $N = 2$, $R = 3$ case, the sparsity assumption, A4 is required. The rationale is that if only a single source is active in M consecutive frames, the local dimensionality of the PARAFAC model will be $R = 1$, and the corresponding column of $\mathbf{A}_k$ is available through equation (9). When estimating $\mathbf{a}_k$ across time, the estimates should ideally occupy R discrete points in space pertaining to the R columns of $\mathbf{A}_k$. In a realistic setting, model bias and slight violations of A4 will cause an amount of dispersion of the estimated $\mathbf{a}_k$ around the true values. Provided this effect is not too severe, $\mathbf{A}_k$ can still be estimated by means of a clustering algorithm. The dispersion around the cluster centers can be quantified by computing the within-class variance, Q.

After $\mathbf{A}_k$ has been acquired through the clustering, $\mathbf{\Lambda}_k$ can be estimated via equation (8) from all frames under less strict conditions on R and N. The typical rank of $(\mathbf{A} \odot \mathbf{A}^*)$ is $\min\left\{\frac{1}{2}N(N+1), R\right\}$. Therefore, $\mathbf{\Lambda}_k$ generally has a unique solution if $\frac{1}{2}N(N+1) \geq R$. As a consequence, the source spectrograms can be recovered when $N = 2$, $R = 3$, provided $\mathbf{A}_k$ is known or has been estimated sufficiently accurately. The identifiability of $\mathbf{\Lambda}_k$ including or excluding sparsity is summarized in table 1.

- discrete Fourier transform Hann-windowed data
- prewhiten
- for all frequencies k:
  - for all blocks of frames $\{n, n+1, \ldots, n+M-1\}$:
    - fit 1-component PARAFAC model
  - estimate $\mathbf{A}_k$ via k-means clustering of $\mathbf{a}_k$ estimates
  - compute $\mathbf{\Lambda}_k$ from $\hat{\mathbf{A}}_k$
  - compute MAP estimate of $\mathbf{s}_k$ from $\mathbf{A}_k$ and $\mathbf{\Lambda}_k$
- solve permutation problem using power corr. across freq.
- inverse prewhiting
- reconstruct time-domain signal by IDFT and overlap-add

Table 2. The blind source separation algorithm for underdetermined convolutive mixtures of sources that are sparse in the time-frequency plane.

6. SOURCE RECONSTRUCTION

Once the $\mathbf{A}_k$ and $\mathbf{D}_k^n$ have been estimated, the sources can be inferred from the data and the parameters of the model. A maximum posteriori (MAP) scheme, which builds on the joint Gaussianity of $\mathbf{s}_t$ and $\mathbf{x}_t$ and the assumptions A1-A3, is employed as suggested in [3]:

$$\hat{\mathbf{s}}_k \quad = \quad \mathbf{D}_k^{(n)}\mathbf{A}_k\left(\mathbf{A}_k\mathbf{D}_k^{(n)}\mathbf{A}_k^H\right)^{-1}\mathbf{x}_k$$

In order to cancel the effects of arbitrary scaling of $\mathbf{A}_k$ and $\mathbf{D}_k^n$, the sources are computed as they appear at the sensors, e.g. the j'th source at the i'th sensor: $(\mathbf{A}_k)_{ij}\hat{s}_{j,k}$. Where the number of sources and sensors are locally equal in the time-frequency plane, the above simply reduces to $\mathbf{s}_k = \mathbf{A}_k^{-1}\mathbf{x}_k$. In case of underdetermined mixtures, the degree of success of the MAP estimation depends on the sparsity assumption, A4. Reconstruction in the time domain is performed by inverse DFT and overlap-add.

7. ALGORITHM

The full set of assumptions, A1-A4, are included in the presented algorithm. A number of implementational issues remains. A preprocessing step is included to better condition the data for clustering and separation. In a band of frequencies a whitening matrix is applied to $\mathbf{x}_k$, i.e. $\tilde{\mathbf{x}}_k = \mathbf{W}\mathbf{x}_k$, so that $\langle\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^H\rangle = \mathbf{I}$, where averaging is over a suitable bandwidth and the signal length. The clustering is carried out in the polar coordinates of $\mathbf{a}$ using k-means. This requires the distance measure to be able to handle circularity. To be more resilient against outliers and violations of the sparsity assumption, the *median* rather than the mean was taken as the cluster center.

The problem of permutation across frequencies was solved by iteratively growing a set in which the permutations are corrected. The measure of similarity is correlation of amplitude as in e.g. [13]. Starting from the highest frequency index, $k = K-1$, to the lowest, $k = 0$, the permutation matrix $\mathbf{P}_k$ was chosen so that it (greedily) maximizes the correlation coefficients of

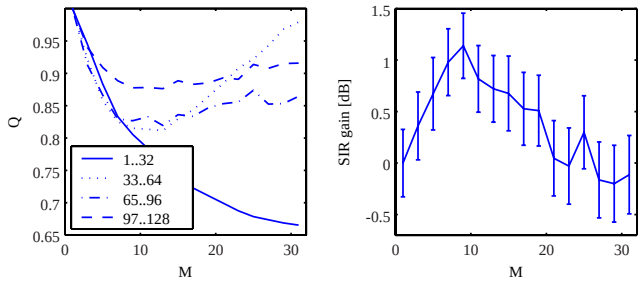

Fig. 2. Effects of the block length M on the clustering of $\mathbf{a}_k$. Left: Within-cluster variance, Q, in 4 frequency bands. Right: The signal to interference (SIR) gain as a function of the segment length M. The results were obtained by averaging across the sources of 3 underdetermined mixtures, $K = 256$.

	$K = 256$	$K = 512$	Araki et al.
mmf	12, 13, 11	16, 15, 10	14, 14, 7
mmm	12, 7, 15	15, 8, 16	11, 4, 14
fff	10, 11, 16	8, 9, 15	5, 18, 18

Table 3. Estimated signal to interference (SIR) ratios for $R = 3$, $N = 2$ mixtures in a simulated reverberant room. f's and m's represent male and female speakers in the mixture.

$|\mathbf{P}_k\hat{\mathbf{s}}_k|$ to $\sum_{k'=k+1}^{K-1}|\mathbf{P}_{k'}\hat{\mathbf{s}}_{k'}|$. This simple method proved fairly robust, and better results were not obtained with the approach of [13]. The algorithm is summarized in table 2.

8. RESULTS AND DISCUSSION

For the initial simulations, experimental audio data generated by researchers Araki et al., see e.g. [10], was used.[3] In each of the mixtures, $R = 3$ speech signals, sample rate $F_s = 8kHz$, were convolved with real room impulse functions of length $\tau_r = 130ms$ to construct $N = 2$ mixtures. The microphones were situated $d_m = 4cm$ apart, the distance to the speakers was $d_s = 110cm$ and the angles to the microphones were $\theta_1 = 50°$, $\theta_2 = 100°$ and $\theta_3 = 135°$. The room dimensions were $\approx 4m \times 4m \times 3m$. Mixture fragments of length $T = 7s$ were used. To measure the degree of separation, the signal-to-interference ratio (SIR) quality index was computed. The SIR's were estimated from the original and estimated sources by means of the `BSS_EVAL` toolbox, see [16] for definitions, and provided a reasonable correlate with human subjective evaluation.[4] Using the sparse algorithm, results on mixtures of same-sex and mixed-sex speech were obtained, see table 3. For reference, the quality measures were also computed for the audio files of [10]. The new method appears to exhibit similar performance to the reference. These results were replicated in a real office environment - 3 male speakers reading aloud from Hans Christian Andersen's fairy tales were separated using the algorithm. The room measures $4.25m \times 5.82m \times 3.28m$. The values of d_m, d_s were left unchanged from the generated mixtures. The estimated source signals can be appreciated at www.imm.dtu.dk/∼rko/underdetermined/index.htm.

[3]The audio wave files are available at www.kecl.ntt.co.jp/icl/signal/araki/nbficademo.html.

[4]The toolbox is downloadable at www.irisa.fr/metiss/bss_eval/. The script `bss_decomp_filt.m` was used with $L = 100$.

Furthermore, the effect of the block length M was assessed, using the same convolutive mixtures, $K = 256$. It is seen in figure 2 that the within-class variance Q at most frequencies approximately peaks at $M \approx 10$. The SIR index has a similar optimum, suggesting that Q can be used to determine the stationarity properties of the signal and select an optimal model order.

9. REFERENCES

[1] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley, 2001.

[2] J. Cardoso, "Blind signal separation: statistical principles," in *IEEE, Blind Identification and Estimation*, 1998, vol. 90.

[3] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Transactions, Speech and Audio Processing*, pp. 320–7, 5 2000.

[4] T.W. Lee, A. J. Bell, and R. H. Lambert, "Blind separation of delayed and convolved sources," in *Advances in NIPS*. 1997, vol. 9, p. 758, The MIT Press.

[5] R. K. Olsson and L. K. Hansen, "A harmonic excitation state-space approach to blind separation of speech," in *Advances in NIPS 17*, 2005, pp. 993–1000.

[6] T. Lee, M. Lewicki, M. Girolami, and T. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representations," *IEEE Signal Processing Letters*, vol. 6, no. 4, 1999.

[7] P. Comon, "Blind identification and source separation in 2x3 under-determined mixtures," *IEEE Transactions on Signal Processing*, vol. 52, pp. 11–22, 2004.

[8] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, pp. 2353–2362, 2001.

[9] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, pp. 1830–1847, 2004.

[10] S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada, "Blind separation of more speakers than sensors with less distortion by combining sparseness and ICA," in *IWAENC2003*, 2003, pp. 271–274.

[11] J. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra and its Applications*, vol. 18, pp. 95–138, 1977.

[12] A. Smilde, R Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, Wiley, 2004.

[13] K. Rahbar and J. P. Reilly, "A new frequency domain method for blind source separation of convolutive audio mixtures," *IEEE Transactions on Speech and Audio Processing*, to appear.

[14] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. on Signal Processing*, , no. 2, pp. 320–7, 1997.

[15] J. M. F. ten Berge and N. D. Sidiropoulos, "On uniqueness in CANDECOMP / PARAFAC," *Psychometrika*, vol. 67, no. 3, pp. 399–409, 2002.

[16] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL Toolbox User Guide," Tech. Rep. 1706, IRISA, Rennes, France, 2005.

Bibliography

Acernese, F., Ciaramella, A., Martino, S. D., Rosa, R. D., Falanga, M., and Tagliaferris, R. (2003). Neural networks for blind-source separation of stromboli explosion quakes. *IEEE Transactions on Neural Networks*, 14(1):167–175.

Anemüller, J. and Kollmeier, B. (2000). Amplitude modulation decorrelation for convolutive blind source separation. In *International Conference on Independent Component Analysis and Blind Signal Separation*, pages 215–220.

Araki, S., Makino, S., Blin, A., Mukai, R., and Sawada, H. (2003). Blind separation of more speakers than sensors with less distortion by combining sparseness and ICA. In *International Workshop on Acoustic Echo and Noise Control*, pages 271–274.

Asari, H., Olsson, R. K., Pearlmutter, B. A., and Zador, A. M. (2007). Sparsification for monaural source separation. In Sawada, H., Araki, S., and Makino, S., editors, *Blind Speech Separation - in press*. Springer.

Asari, H., Pearlmutter, B. A., and Zador, A. M. (2006). Sparse representations for the cocktail party problem. *Journal of Neuroscience*, 26(28):7477–90.

Attias, H. and Schreiner, C. E. (1998). Blind source separation and deconvolution: the dynamic component analysis algorithm. *Neural Computation*, 10(6):1373–1424.

Bach, F. R. and Jordan, M. I. (2005). Blind one-microphone speech separation. In *Advances in Neural Information Processing Systems*.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

Benaroya, L., Donagh, L. M., Bimbot, F., and Gribonval, R. (2003). Non negative sparse representation for wiener based source separation with a single sensor. In

International Conference on Acoustics, Speech, and Signal Processing, pages 613–616.

Bofill, P. and Zibulevsky, M. (2001). Underdetermined blind source separation using sparse representations. *Signal Processing*, 81:2353–2362.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120.

Bregman, A. S. (1990). *Auditory Scene Analysis: the perceptual organisation of sound*. The MIT Press, Cambridge, Massachusetts, London, England.

Cardoso, J.-F., H. Snoussi, J. D., and Patanchon, G. (2002). Blind separation of noisy gaussian stationary sources. application to cosmic microwave background imaging. In *European Signal Processing Conference*, pages 561–564.

Casey, M. and Westner, A. (2000). Separation of mixed audio sources by independent subspace analysis. In *International Computer Music Conference*.

Cauwenberghs, G. (1999). Monaural separation of independent acoustical components. In *IEEE Int. Symp. Circuits and Systems*, pages 62–65.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *Journal of Acoustic Society of America*, 25:975–979.

Chiappa, S. and Barber, D. (2005). Baysian linear state space models for biosignal decomposition. Technical Report Technical Report IDIAP-RR 05-84, IDIAP.

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

Cooke, M. P., Barker, J., Cunningham, S. P., and Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *Journal of the Acoustical Society of America*, 120:2421–2424.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistics Society, Series B*, 39:1–38.

Dyrholm, M., Makeig, S., and Hansen, L. K. (2007). Model selection for convolutive ica with an application to spatiotemporal analysis of eeg. *Neural Computation*, 19:934–955.

Eggert, J. and Körner, E. (2004). Sparse coding and NMF. In *IEEE International Conference on Neural Networks*, volume 4, pages 2529–2533.

Ellis, D. (2004). Evaluating speech separation systems. In Divenyi, P., editor, *Speech Separation by Humans and Machines*. Springer.

Gardner, T. J. and Magnasco, M. O. (2006). Sparse time-frequency representations. *Proceedings National Academy of Sciences of the United States of America*, 103(16):6094–9.

Gerven, S. V. and Compernolle, D. V. (1995). Signal separation by symetric adaptive decorrelation: Stability, convergence, and uniqueness. *IEEE Trans. Signal Processing*, 43:1602–1612.

Højen-Sørensen, P. A., Winther, O., and Hansen, L. K. (2002). Mean field approaches to independent component analysis. *Neural Computation*, 14:889–918.

Hu, G. and Wang, D. L. (2003). Monaural speech separation. In *Advances in Neural Information Processing Systems*.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.

Jang, G. J. and Lee, T. W. (2003). A maximum likelihood approach to single channel source separation. *Journal of Machine Learning Research*, 4:1365–1392.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45.

Krim, H. and Viberg, M. (1996). Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13(4):67–94.

Kristjansson, T., Attias, H., and Hershey, J. (2004). Single microphone source separation using high resolution signal reconstruction. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 817–821.

Kristjansson, T., Hershey, J., Olsen, P., Rennie, S., and Gopinath, R. (2006). Super-human multi-talker speech recognition. In *International Conference on Spoken Language Processing*.

Kruskal, J. (1977). Three-way arrays: Rank and uniqueness of trilinear decompo-

sitions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18:95–138.

Lee, T., Bell, A. J., and Lambert, R. H. (1997). Blind separation of delayed and convolved sources. In *Advances in Neural Information Processing Systems*, volume 9, page 758.

MacKay, D. (1996). Maximum likelihood and covariant algorithms for independent component analysis. Available from http://wol.ra.phy.cam.ac.uk/mackay/.

McAulay, R. and Quateri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(4):744–754.

McKeown, M., Hansen, L. K., and Sejnowski, T. J. (2003). Independent component analysis for fmri: What is signal and what is noise? *Current Opinion in Neurobiology*, 13(5):620–629.

Michie, D., Spiegelhalter, D., and (eds), C. T. (1994). *Machine Learning, Neural and Statistical Classification*. http://www.amsta.leeds.ac.uk/ charles/statlog/indexdos.html.

Mitianoudis, N. (2004). *Audio Source Separation using Independent Component Analysis*. PhD thesis, Queen Mary, University of London.

Molgedey, L. and Schuster, G. (1994). Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637.

Moulines, E., Cardoso, J., and Cassiat, E. (1997). Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 3617–3620.

Murata, N., Ikeda, S., and Ziehe, A. (2001). An approach to blind source separation based on temporal structure of speech signals. *Neurocomputing*, 41(1-4):1–24.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Olsson, R. K. and Hansen, L. K. (2004a). Estimating the number of sources in a

noisy convolutive mixture using BIC. In *International Conference on Independent Component Analysis and Blind Signal Separation*, pages 618–625.

Olsson, R. K. and Hansen, L. K. (2004b). Probabilistic blind deconvolution of non-stationary sources. In *European Signal Processing Conference*, pages 1697–1700.

Olsson, R. K. and Hansen, L. K. (2005). A harmonic excitation state-space approach to blind separation of speech. In *Advances in Neural Information Processing Systems*, volume 17, pages 993–1000.

Olsson, R. K. and Hansen, L. K. (2006a). Blind separation of more sources than sensors in convolutive mixtures. In *International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 657–660.

Olsson, R. K. and Hansen, L. K. (2006b). Linear state-space models for blind source. *Journal of Machine Learning Research*, 7:2585–2602.

Olsson, R. K., Petersen, K. B., and Lehn-Schiøler, T. (2007). State-space models - from the EM algorithm to a gradient approach. *Neural Computation*, 19(4):1097–1111.

Parra, L. and Sajda, P. (2003). Blind source separation via generalized eigenvalue decomposition. *Journal of Machine Learning Research*, 4:1261–1269.

Parra, L. and Spence, C. (2000). Convolutive blind separation of non-stationary sources. *IEEE Transactions, Speech and Audio Processing*, pages 320–7.

Pearlmutter, B. A. and Olsson, R. K. (2006). Algorithmic differentiation of linear programs for single-channel source separation. In *IEEE International Workshop on Machine Learning and Signal Processing*.

Pearlmutter, B. A. and Parra, L. C. (1997). A context-sensitive generalization of ICA. In *Advances in Neural Information Processing Systems*, volume 9, pages 613–619.

Pedersen, M. S., Larsen, J., Kjems, U., and Parra, L. C. (2007). A survey of convolutive blind source separation methods. In *Springer Handbook of Speech (to appear)*. Springer Press.

Pontoppidan, N. H. (2006). *Condition Monitoring and Management from Acoustic Emissions*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rahbar, K. and Reilly, J. P. (2005). A frequency domain method for blind source separation of convolutive audio mixtures. *IEEE Transactions on Speech and Audio Processing*, 13(5):832–844.

Rauch, H. E., Tung, F., and Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *American Institute of Aeronautics and Astronautics*, 3(8):1445–50.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–7.

Roman, N., Wang, D., and Brown, G. J. (2004). A classification-based cocktail-party processor. In *Advances in Neural Information Processing Systems*, volume 16.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345.

Roweis, S. T. (2001). One microphone source separation. In *Advances in Neural Information Processing Systems*, pages 793–799.

Roweis, S. T. (2003). Factorial models and refiltering for speech separation and denoising. In *Eurospeech*, pages 1009–1012.

Särelä, J. (2004). *Eploratory Source Separation In Biomedical Systems*. PhD thesis, Helsinki University of Technology.

Schmidt, M. N., Larsen, J., and Hsiao, F. (2007). Wind noise reduction using sparse non-negative matrix factorization. In *IEEE International Workshop on Machine Learning for Signal Processing*.

Schmidt, M. N. and Olsson, R. K. (2006). Single-channel speech separation using sparse non-negative matrix factorization. In *International Conference on Spoken Language Processing*.

Schmidt, M. N. and Olsson, R. K. (2007). Feature space reconstruction for single-channel speech separation. In *Workshop on Applications of Signal Processing to Audio and Acoustics*.

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

Smaragdis, P. (2004). Discovering auditory objects through non-negativity constraints. In *Statistical and Perceptual Audio Processing*.

Smaragdis, P. (2007). Convolutive speech bases and their application to supervised speech separation. *IEEE Transaction on Audio, Speech and Language Processing - to appear*.

Smilde, A., Bro, R., and Geladi, P. (2004). *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley.

Stevens, S. S. (1957). On the psychophysical law. *Psychological Review*, 64(3):153–181.

Thi, H. N. and Jutten, C. (1995). Blind source separation for convolutive mixtures. *Signal Processing*, 45:209–229.

Torkkola, K. (1999). Blind separation for audio signals are we there yet? In *International Conference on Independent Component Analysis and Blind Signal Separation*, page 239244.

Virtanen, T. (2006a). *Sound Source Separation in Monaural Music Signals*. PhD thesis, Tampere University of Technology.

Virtanen, T. (2006b). Speech recognition using factorial hidden markov models for separation in the feature space. In *International Conference on Spoken Language Processing*.

Wang, D. L. and Brown, G. J. (1999). Separation of speech from interfering sounds based on oscillatory correlation. *IEEE Transactions on Neural Networks*, 10(3):684.

Weinstein, E., Feder, M., and Oppenheim, A. V. (1993). Multi-channel signal separation by decorrelation. *IEEE transactions on speech and audio processing*, 1(4).

Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, New York.

Yellin, D. and Weinstein, E. (1996). Multichannel signal separation: Methods and analysis. *IEEE Transactions on signal processing*, 44(1):106–118.

Yilmaz, O. and Rickard, S. (2004). Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52:1830–1847.

Zhang, J., Khor, L. C., Woo, W. L., and Dlay, S. (2006). A maximum likelihood approach to nonlinear convolutive blind source separation. In *International*

Conference on Independent Component Analysis and Blind Signal Separation, pages 926–33.